

**UNIVERSIDAD CARLOS III DE MADRID**  
**ESCUELA POLITÉCNICA SUPERIOR**



TRABAJO DE FIN DE GRADO

Grado de Ingeniería en Tecnologías Industriales

**ANÁLISIS COMPARATIVO DE LOS PROTOCOLOS EXISTENTES  
PARA LOS ENSAYOS DE INTEROPERABILIDAD (IOP  
TESTING): SUBSET-112 Y RAILML**

Realizado por:

**Ainara Matey Benito**

Dirigido por:

**María Malfaz Vázquez**

**Raquel Martínez Barbero**

Curso 2016/2017 - Convocatoria Julio

## AGRADECIMIENTOS

---

Habiendo dado por finalizado este proyecto, me gustaría agradecer a las personas e instituciones que han hecho posible que este proyecto haya salido adelante tras el esfuerzo realizado.

A Raquel Martínez Barbero, por darme esta oportunidad y permitirme trabajar con ella y ayudarme como tutora tanto en el transcurso de mis prácticas en ADIF como en el desarrollo de mi Trabajo de Fin de Grado.

A Daniel Molina Marinas por haberme brindado la oportunidad de ayudar al CEDEX en la colaboración del proyecto VITE, así como por haberme ofrecido instrucciones para mejorar mi proyecto.

A María Malfaz Vázquez, por su implicación y ayuda desde la Universidad con el desarrollo de la memoria del proyecto.

Por otra parte me gustaría agradecer a las instituciones implicadas, ADIF, CEDEX y la Universidad Carlos III de Madrid, por hacer posible la realización de este Trabajo de Fin de Grado.

Finalmente, me gustaría citar a mis padres, por su apoyo constante y su fe en mí, motivándome día a día. Su apoyo ha sido fundamental para mí, no solo en este proyecto, sino en todos los aspectos de mi vida.

Muchas gracias por todo

Ainara



## CONTENIDO

---

### ÍNDICE

<b>AGRADECIMIENTOS .....</b>	<b>2</b>
<b>CONTENIDO .....</b>	<b>3</b>
ÍNDICE .....	3
ÍNDICE DE FIGURAS.....	6
ÍNDICE DE TABLAS .....	8
<b>RESUMEN.....</b>	<b>9</b>
<b>ABSTRACT .....</b>	<b>11</b>
<b>1 INTRODUCCIÓN.....</b>	<b>13</b>
<b>2 CONTEXTO .....</b>	<b>14</b>
2.1 CONTEXTO EUROPEO Y PROYECTO SHIFT2RAIL.....	14
2.2 IP2 Y VITE .....	16
2.3 OBJETIVO DEL TRABAJO DE FIN DE GRADO: ENSAYOS DE INTEROPERABILIDAD.....	19
<b>3 OBJETIVOS DEL TRABAJO DE FIN DE GRADO .....</b>	<b>21</b>
<b>4 ESTRUCTURA DEL DOCUMENTO .....</b>	<b>22</b>
<b>5 ESTADO DEL ARTE .....</b>	<b>23</b>
5.1 ERTMS.....	25
5.2 ENSAYOS DE INTEROPERABILIDAD ERTMS/ETCS .....	27
<b>6 ESPECIFICACIONES TÉCNICAS.....</b>	<b>29</b>
6.1 XML SCHEMA DEFINITION.....	30
6.2 FORMATOS DE DATOS ERTMS.....	32
6.2.1 UNISIG (Subset-112).....	32
6.2.2 RailML.....	34
<b>7 ANÁLISIS COMPARATIVO.....</b>	<b>36</b>
7.1 UNISIG (SUBSET-112) .....	36
7.2 RAILML.....	38



7.3	ELEMENTOS TOPOLÓGICOS.....	40
7.3.1	Topología de UNISIG (Subset-112).....	40
7.3.2	Topología de RailML.....	43
7.3.3	Traducción de la topología de Subset-112 a RailML.....	45
7.3.3.1	Conexión de dos segmentos.....	46
7.3.3.2	Final de vía.....	48
7.3.3.3	Conexión externa.....	50
7.3.3.4	Conexión entre tres segmentos (aguja).....	53
7.3.3.5	Cruces, travesías de unión simple y travesías de unión doble.....	56
7.3.3.6	Descarrilador.....	59
7.4	POSICIONAMIENTO Y DIRECCIÓN.....	62
7.4.1	UNISIG (Subset-112).....	62
7.4.2	RailML.....	65
7.4.3	Traducción del posicionamiento Subset-112 a RailML.....	66
7.4.4	Traducción de la dirección.....	67
7.5	ELEMENTOS EN LA VÍA.....	69
7.5.1	Elementos físicos.....	70
7.5.1.1	Cambio de gálibo.....	70
7.5.1.2	Cambios en el voltaje.....	72
7.5.1.3	Cambio de categoría de carga por eje.....	74
7.5.1.4	Cambios estáticos de velocidad.....	77
7.5.1.5	Cambios de pendiente.....	79
7.5.1.6	Condiciones de la vía.....	80
7.5.1.7	Cambios de límites de corriente.....	82
7.5.1.8	Pasos a nivel.....	83
7.5.1.9	Andenes.....	85
7.5.1.10	Áreas de marcha atrás.....	87
7.5.2	Sistemas de operación o control.....	88
7.5.2.1	Señales.....	89
7.5.2.2	Balizas.....	90
7.5.2.3	Eurolazos y RIUs.....	93
7.5.2.4	Ocupación de la vía.....	96
7.5.2.5	Sistemas NTC.....	98
7.5.3	Puntos de operación o control.....	100
7.5.3.1	Estaciones.....	101
7.5.3.2	Áreas de maniobras.....	101
7.5.4	Atributos de la infraestructura.....	103



7.5.4.1	Valores nacionales.....	103
7.5.4.2	Áreas de ETCS.....	109
7.6	PROPIEDADES DE LA INFRAESTRUCTURA .....	112
7.7	RUTAS.....	113
7.8	VISUALIZACIÓN .....	116
<b>8</b>	<b>PRESUPUESTO DEL PROYECTO .....</b>	<b>117</b>
<b>9</b>	<b>CONCLUSIONES Y PROPUESTAS .....</b>	<b>118</b>
9.1	PROPUESTAS DE MEJORA .....	118
9.1.1	UNISIG SS 112.....	118
9.1.2	RailML.....	119
9.2	CONCLUSIONES GENERALES.....	120
<b>10</b>	<b>GLOSARIO DE TÉRMINOS .....</b>	<b>122</b>
<b>11</b>	<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>123</b>

## ÍNDICE DE FIGURAS

FIGURA 1. INTERDEPENDENCIA DE WPS .....	19
FIGURA 2: SISTEMAS ATP NACIONALES EN EUROPA[7] .....	24
FIGURA 3:IMPLEMENTACIÓN DE ERTMS A NIVEL MUNDIAL[9] .....	27
FIGURA 4: LABORATORIO DE INTEROPERABILIDAD FERROVIARIA [5].....	28
FIGURA 5: ESQUEMA RAILML [13].....	31
FIGURA 6: LOGO DE UNISIG.....	32
FIGURA 7: LOGO DE RAILML .....	34
FIGURA 8: ESTRUCTURA BÁSICA RAILML. ....	39
FIGURA 9:EJEMPLO DE TOPOLOGÍA EN SUBSET 112 .....	40
FIGURA 10: TOPE DE FINAL DE VÍA (BUFFER STOP) EN LA ESTACIÓN DE BARCELONA [8].....	43
FIGURA 11: AGUJA EN LA ESTACIÓN DE VILLA JOYOSA [8] .....	44
FIGURA 12: ANTIGUO CRUZAMIENTO EN PALENCIA [8].....	45
FIGURA 13: CAMBIO KILOMÉTRICO .....	46
FIGURA 14: EJEMPLO DE CONEXIÓN ENTRE DOS SEGMENTOS EN SUBSET-112 .....	47
FIGURA 15: EJEMPLO DE CONEXIÓN ENTRE DOS SEGMENTOS EN RAILML.....	47
FIGURA 17: EJEMPLO DE FINAL DE VÍA EN SUBSET-112.....	49
FIGURA 16: FIN DE VÍA .....	49
FIGURA 18: EJEMPLO DE FINAL DE VÍA EN RAILML.....	50
FIGURA 19: CONEXIÓN EXTERNA EN SUBSET-112.....	51
FIGURA 20: CONEXIÓN EXTERNA (OPEN END) EN SUBSET 112 .....	52
FIGURA 21: CONEXIÓN EXTERNA (CONNECTION) EN SUBSET 112 .....	52
FIGURA 22: AGUJA .....	53
FIGURA 23: AGUJA (PUNTO) EN SUBSET-112.....	54
FIGURA 24:AGUJA (PUNTO) EN RAILML .....	55
FIGURA 25 CRUZAMIENTO, TRAVESÍA DE DOBLE UNIÓN Y DE SIMPLE UNIÓN EN SUBSET 112.....	56
FIGURA 26: CRUZAMIENTO (1),TRAVESÍA DE DOBLE UNIÓN (2) Y DE SIMPLE UNIÓN (3) EN SUBSET-112 .....	57
FIGURA 27: TRAVESÍA O CRUZAMIENTO EN RAILML .....	58
FIGURA 28: DESCARRILADOR .....	59
FIGURA 29: DESCARRILADOR EN SUBSET 112 .....	60
FIGURA 30:DESCARRILADOR EN RAILML .....	61
FIGURA 31: POSICIONAMIENTO PUNTUAL EN SUBSET-112 .....	63
FIGURA 32: POSICIONAMIENTO DE UNA LÍNEA EN SUBSET-112 .....	64
FIGURA 33: POSICIONAMIENTO DE UN ÁREA EN SUBSET-112 .....	64
FIGURA 34: POSICIONAMIENTO PUNTUAL .....	66
FIGURA 35: POSICIONAMIENTO DE UN ELEMENTO CON LONGITUD.....	67
FIGURA 36: ELEMENTOS DE VÍA .....	69
FIGURA 37: GÁLIBOS INTERNACIONALES.[19].....	71



FIGURA 38: VOLTAJES USADOS EN EUROPA (SIMPLIFICADO)[2] .....	73
FIGURA 39: EJEMPLO DE PENDIENTES .....	80
FIGURA 41: ESQUEMA DE LA EXTENSIÓN DE RAILML PARA LEVELCROSSING.....	84
FIGURA 40: PASO A NIVEL .....	84
FIGURA 42: ZONAS EN UN PASO A NIVEL .....	85
FIGURA 43: ANDÉN.....	85
FIGURA 44: ALTURA DEL ANDÉN .....	86
FIGURA 45: ESQUEMA DE LA EXTENSIÓN DE RAILML PARA REVESINGAREA .....	88
FIGURA 46: EJEMPLOS DE SEÑALES FERROVIARIAS [19] .....	89
FIGURA 47: BALIZA ERTMS [8] .....	90
FIGURA 48: ESTRUCTURA ELEMENTO BALIZA.....	91
FIGURA 49:ESQUEMA DE LA EXTENSIÓN DE RAILML PARA STATE, DENTRO DE BALISE.....	93
FIGURA 50: ESTRUCTURA EUROLAZO.....	94
FIGURA 51: ESTRUCTURA RIU .....	95
FIGURA 52:ESQUEMA DE LA EXTENSIÓN DE RAILML PARA RIU .....	95
FIGURA 53: ESQUEMA DE LA EXTENSIÓN DE RAILML PARA EUROLOOP .....	96
FIGURA 54: CIRCUITO DE VÍA .....	97
FIGURA 55: ESTRUCTURA DISPOSITIVO NTC .....	98
FIGURA 56: ESTRUCTURA LAZO NTC.....	99
FIGURA 57:ESQUEMA DE LA EXTENSIÓN DE RAILML PARA NTCLOOP.....	100
FIGURA 58: ESQUEMA DE LA EXTENSIÓN DE RAILML PARA NTCLOOP.....	100
FIGURA 59: ÁREA DE MANIOBRAS DE BARECELONA-CAN TUNIS[24] .....	102
FIGURA 60: EXTENSIÓN DE RAILML PARA OCP.....	102
FIGURA 61: ESQUEMA DE LA EXTENSIÓN DE RAILML PARA NATIONALVALUECHANGES.....	105
FIGURA 62: ESQUEMA DE EXTENSIÓN DE RAILML PARA ETCSAREA.....	111
FIGURA 63: VISUALIZACIÓN DE VÍA.....	116



## ÍNDICE DE TABLAS

TABLA 1: ESTRUCTURA DEL DESGLOSE EN TIERRA .....	38
TABLA 2: VALORES DIRECTIONENUM .....	68
TABLA 3: GÁLIBOS SUBSET-112 .....	72
TABLA 4: VALORES DE VOLTAJES UNISIG[20] .....	74
TABLA 5: CLASIFICACIÓN DE CATEGORÍAS DE CARGA POR EJE POR UNISIG[20] .....	76
TABLA 6: CATEGORÍAS PARA VELOCIDADES EN RAILML Y UNISIG .....	79
TABLA 7: CORRELACIÓN DE TIPOS DE CONDICIONES DE VÍA .....	82
TABLA 8: VALORES PARA LA ALTURA DE PLATAFORMA EN UNISIG [13] .....	86
TABLA 9: VALORES PARA EL LADO DEL ANDÉN EN UNISIG [13] .....	87
TABLA 10: VELOCIDADES MÁXIMAS DE VALORES NACIONALES .....	106
TABLA 11: VALORES DE RADIOREACTION .....	107
TABLA 12: VALORES DE CONFIDENCELEVELEB .....	108
TABLA 13: VALORES DE LENGTH (KRCORRECTIONS) .....	109
TABLA 14: VALORES DE ETCSLEVEL .....	110
TABLA 15: VALORES ETCSVERSION .....	110
TABLA 16: VALORES DEVICETYPE .....	113
TABLA 17: MODOS OPERATIVOS .....	114
TABLA 18: COSTES .....	117
TABLA 19: GLOSARIO DE TÉRMINOS .....	122



## RESUMEN

---

Hoy en día, la globalización creciente requiere de infraestructuras eficientes y rápidas para el transporte de información, mercancías y personas, circunstancias ante las cuales la industria ferroviaria debe estar a la altura.

Uno de los principales problemas que sufre la industria ferroviaria ante la expansión internacional, es la falta de interoperabilidad entre las infraestructuras, trenes y equipos de diferentes países. Esto provoca retrasos, costes añadidos y dificultades técnicas que impiden el progreso de esta industria y su posicionamiento competitivo frente a otras alternativas como el transporte por carretera.

El plan estratégico que plantea Europa para la innovación ferroviaria, enmarcado en el programa Shift2Rail, pretende dar un vuelco a esta situación mejorando esta industria a todos los niveles mediante la innovación. Dentro de este marco estratégico, se ha consolidado el proyecto VITE, cuyo principal objetivo es unificar los distintos criterios que existen en señalización ferroviaria para poder realizar el máximo número de pruebas de interoperabilidad en los laboratorios con todas las garantías necesarias que permita llevar a niveles mínimos las pruebas en campo o pruebas in situ, ahorrando así grandes costes y reduciendo plazos.

A la hora de intercambiar información sobre los sistemas de señalización para infraestructura y operación ferroviaria, es importante utilizar un formato de datos común para evitar así interpretaciones erróneas de información, lo que se puede traducir en problemas de seguridad por una señalización poco eficaz.

Actualmente no existe un formato estándar único, por lo que se están valorando y desarrollando varios formatos que pretenden hacerse comunes agrupando las necesidades actuales de todas las aplicaciones ferroviarias existentes.

Este documento busca contribuir a esa estandarización, presentando dos de los formatos mejor valorados y más utilizados por la industria ferroviaria, tanto operadores, constructores, ingenierías, laboratorios y administradores. Estos dos formatos son aquellos actualmente ofrecidos por las organizaciones colaborativas de interoperabilidad: UNISIG (Subset-112) y RailML.



El formato desarrollado por UNISIG es el que hasta ahora ha sido empleado en los laboratorios de investigación españoles, mientras que RailML es el formato que están adoptando muchas de las empresas de ingeniería y desarrollo ferroviario y tiene muchas posibilidades de convertirse en el formato estándar para este tipo de aplicaciones ferroviarias en Europa y probablemente en otras partes del mundo.

Es por esto que en el desarrollo de este documento se valoran y se comparan ambos protocolos, realizando una traducción del formato de las especificaciones para el entorno de pruebas ferroviario ofrecido por UNISIG al que RailML está desarrollando, para facilitar a la innovación ferroviaria española adaptarse a proyectos europeos como el VITE.

## ABSTRACT

---

Nowadays, increasing globalization requires fast and efficient infrastructures for transporting information, freights and passengers. The railway industry should face this development and make changes to support more movement across borders with better technology

One of the main problems railway industry is facing in international expansion is the lack of interoperability among infrastructures, trains and equipments from different countries. This leads to delays, increasing costs and technical difficulties. This prevents the industry's progress from reaching a competitive edge with respect to other transporting choices such as road transport.

The strategic plan set out by Europe for railway innovation, included in the project Shift2Rail intends to completely shift this situation improving railway transport in every aspect with research and innovation. In this project, a key subproject is VITE. VITE is intended to unify the signaling systems and move as many interoperability tests as possible to the laboratories. This is to reach the Zero Onsite Testing goal, saving great amounts in costs and reducing the time spent in projects.

When it comes to exchanging data among foreign signaling systems concerning information about infrastructure and train operation, it is of great importance to use a common data format. This common format is needed to avoid misleading information and consequent errors. These mistakes could lead to security issues due to signaling problems.

Currently, such a standard format does not exist. This is the reason why some formats are being developed trying to gather all the existing railway applications' needs to become a Europe-wide standard.

This document seeks to contribute to that standardization, presenting two of the best ranked and most used data formats inside the railway industry, including operators, engineering, constructors, laboratories and managers. These two formats are those currently offered by the collaborative interoperability organizations: UNISIG (Subset-112) and RailML.



The data format developed by UNISIG is the one which has been more commonly used in Spanish research facilities. While the RailML format, which is currently under development has increasing chances of becoming the standard data format for railway data exchange applications, not only in Europe, but also in many other places around the globe.

In this document, both protocols are analyzed and compared, translating the format for the specifications for the testing environment offered by UNISIG to the specifications RailML is developing. This way, it will be easier for the Spanish railway industry to adapt to European projects such as VITE.

# 1 INTRODUCCIÓN

---

Este TFG ha sido desarrollado durante la realización de las prácticas externas en ADIF (Administrador de Infraestructuras Ferroviarias), en la Jefatura de Área de Proyectos de I+D+i. Es por esto que el proyecto, al estar relacionado con los proyectos en los que ADIF está involucrado, está estrechamente vinculado con el ámbito ferroviario.

El trabajo está enmarcado en el contexto de un proyecto de innovación europeo en el sector ferroviario, buscando incrementar la relevancia de este sector en el mundo del transporte.

A continuación, se presenta el marco de trabajo en donde se ha desarrollado este TFG, el marco europeo en el que se encuadra y, en concreto el proyecto del que ha formado parte.

Posteriormente, se exponen los objetivos del proyecto y la estructura en la que estos se van a desarrollar.

Después, se presenta el desarrollo del proyecto, mostrando la explicación de todo el trabajo realizado como aportación de este proyecto al sector ferroviario.

Finalmente, se exponen las conclusiones y propuestas de mejora para el desarrollo del mismo.

## 2 CONTEXTO

---

### 2.1 CONTEXTO EUROPEO Y PROYECTO SHIFT2RAIL

En el transcurso de los últimos años, se ha producido una gran evolución y progreso global a nivel social, cultural, económico y político, liderado fundamentalmente por la globalización, fomentada por la capacidad de intercambios tanto de información, productos y personas con cualquier punto del mundo.

Para poder continuar con este progreso, resulta esencial mejorar estas conexiones para facilitar así las relaciones humanas a todos los niveles. Es necesario avanzar e incrementar la calidad de las infraestructuras básicas de comunicación y transporte, tanto de mercancías como de personas, ayudando así al desarrollo socioeconómico de la población.

La mejora de la competitividad en las infraestructuras de transporte, como puede ser en un sector tan grande como el ferroviario, pasa por un aumento de su capacidad para satisfacer una demanda de tráfico cada vez más creciente, a la vez que su eficacia, consiguiendo una mayor rapidez y puntualidad que vaya acorde con un mundo en el que la inmediatez supone un rasgo cada vez más predominante. Por otra parte, resulta importante mejorar el compromiso de la industria con la sostenibilidad, ya que creando una oferta más atractiva y un aumento de la demanda en el sector ferroviario sobre el tráfico en carretera, disminuirían las emisiones perjudiciales. También es crucial tener una elevada eficiencia en la utilización de los recursos disponibles, centrándose en la mejora en la administración de costes, reduciéndolos en la medida de lo posible, ya que la mayoría de proyectos de innovación en el sector dependen de fondos públicos. Para esto resulta esencial simplificar todo el proceso de negocio, desarrollando una estandarización común y una mayor interoperabilidad entre diferentes sectores y mercados.

Dentro del contexto europeo, se contemplan todos estos objetivos para mejorar la competitividad en el sector con respecto a otros organismos emergentes que están poniéndose en cabeza en innovación y desarrollo (I+D). También se tratan otras

cuestiones internas relacionadas con la organización y administración del sistema ferroviario, con el objetivo de hacerlo más atractivo y eficiente.

Para alcanzar dichas metas, la Comisión Europea (CE) puso en marcha en 2014 el programa de innovación Horizonte 2020 (H2020), bajo el que se creó el acuerdo de colaboración público-privado (Public-Private-Partnership, PPP) denominado **Shift2Rail Joint Undertaking** (S2R JU) [1]. Este programa proporciona una plataforma de actuación a sus colaboradores para la implementación de una estrategia conjunta de innovación, así como un marco europeo común de actuación, haciendo un mejor uso de todos los recursos disponibles como las infraestructuras y el capital humano, así como de las inversiones recibidas, para que de este modo resulten ser menos arriesgadas. Otro de los objetivos conjuntos es aunar las normativas, tecnologías e infraestructuras existentes, creando la Single European Rail Area (SERA), ya que en múltiples ocasiones las diferencias encontradas entre diversos países dificultan gravemente la interoperabilidad entre los diferentes países europeos.

Entre los colaboradores del S2R JU se encuentra la Unión Europea (UE) así como ocho empresas del panorama ferroviario que comprenden fabricantes de equipamiento de vía como Alstom, Ansaldo STS, Bombardier, Construcciones y Auxiliar de Ferrocarriles (CAF), Siemens y Thales, gestores de infraestructura como Network Rail y Trafikverket, y miembros asociados elegidos mediante una convocatoria abierta.

Con objeto de alcanzar dichos objetivos, se han establecido dentro del plan cinco programas denominados IPs (Innovation Programmes), cada uno dedicado a un campo de actuación diferente, como prosigue:

- **IP1:** Trenes fiables y rentables, incluyendo trenes de alta capacidad y alta velocidad.
- **IP2:** Gestión avanzada de tráfico y sistemas de control.
- **IP3:** Infraestructura de alta capacidad rentable, sostenible y fiable.
- **IP4:** Soluciones TIC para servicios ferroviarios atractivos.
- **IP5:** Tecnologías para Mercancías Europeas sostenibles y atractivas.

Estos programas no suponen marcos de actuación aislados, sino que están altamente interconectados, ya que persiguen los mismos objetivos.

El Trabajo de Fin de Grado desarrollado, que aquí se presenta, forma parte de uno de los programas, el IP2, y, en concreto, dentro de uno de los sub-programas denominado: Virtualisation of the Test Environment (VITE). A continuación, pasamos a describir brevemente los objetivos que se persiguen en dichos programas.

## 2.2 IP2 y VITE

El programa IP2 antes mencionado está dedicado a la gestión y tráfico de trenes, así como a los sistemas de control que lo regulan.

El principal objetivo al cual se ajusta este paquete, es mejorar la competitividad de las soluciones europeas para la mejora de la gestión y control del tráfico ferroviario, consiguiendo ir más allá del enfoque único de la seguridad, y llegando a un control en tiempo real que suponga un aumento significativo en la flexibilidad y frecuencia de trenes, pudiendo aumentar así notablemente la capacidad de la industria ferroviaria.

Una de las tecnologías empleadas con este fin es la tecnología de señalización European Rail Traffic Management System (ERTMS) que es el sistema de señalización y control de tráfico que se pretende implantar en todo el sistema ferroviario europeo, fomentando la SERA y salvando así las diferencias entre tecnologías de diferentes países para mejorar de este modo la interoperabilidad y la libre circulación de trenes entre todos los estados miembros.

Ésta es una tecnología muy conocida y actualmente en uso en el ámbito ferroviario mundial y fundamentalmente europeo, pero todavía no se está empleando en toda su extensión y con todas sus posibilidades [2]. Su futuro éxito está tanto en la reducción de costes como en la simplificación de la implementación de nuevos procedimientos de puesta en servicio, para así aumentar la flexibilidad de procesos de autorización y certificación.

Para la implementación de sistemas ERTMS en trenes, actualmente se requiere realizar un alto número de ensayos *in situ* para testar la interoperabilidad entre los componentes que conforman el ERTMS que son los dispositivos en vía y aquellos que se encuentran a bordo. Estos tests suponen cerrar la vía para la prueba, dejándola sin servicio para otras



circulaciones mientras se realizan dichas pruebas, además de que el traslado del tren a vía y toda la instrumentación y sensorización que se necesita, requieren altas inversiones de tiempo y dinero. Según el Plan de Acción del S2R, los ensayos en vía suponen al menos un 30% del coste total de un proyecto de estas características. Por ejemplo, en las pruebas realizadas para poner en servicio sistemas ERTMS en el tramo Barcelona-Figueras, de 140 km, con 70 jornadas de pruebas, se requirió de un presupuesto de 475.000 €, contando con un tren y dos técnicos por jornada [3]. Por lo tanto, reducir su coste tendría un gran impacto en el coste de un proyecto.

Por lo tanto, una solución a este problema es poder crear una *Virtualización del entorno de pruebas*, es decir, el objetivo principal del proyecto anteriormente mencionado: VITE. La propuesta del VITE pasaría por poder ser capaces de llevar a cabo todas estas pruebas en laboratorio, donde se podrían testar un gran número de situaciones, trenes y sistemas de señalización sin salir a vía, siendo así entre 5 y 10 veces más efectivo que llevando a cabo dichas pruebas *in situ*. El principal obstáculo que presenta este tipo de pruebas es que es necesario demostrar que son económicamente rentables, así como demostrar su fiabilidad, es decir, hasta qué punto los resultados de los ensayos llevados a cabo en laboratorio son consistentes con lo que sucedería en una situación real.

VITE está liderado fundamentalmente por empresas españolas e italianas, además de contar con la colaboración de Bélgica y República Checa. Tanto en España como en Italia se han realizado ya varios ensayos en laboratorio de otros tipos de pruebas previos a la implantación de estos sistemas de señalización, lo que confiere a este proyecto un extra de fiabilidad y una mayor probabilidad de éxito, demostrando así los reducidos riesgos de inversión en esta idea.

Las empresas colaboradoras del proyecto VITE son, junto con la colaboración de algunos componentes de S2R JU[4]:

- España:
  - Administrador De Infraestructuras Ferroviarias (ADIF)
  - Ingeniería y economía del transporte S.A. (INECO)
  - Centro de Estudios y Experimentación de Obras Públicas (CEDEX)
  - Renfe-Fabricación y Mantenimiento S.A. (RENFE)
  - Asociación de Acción Ferroviaria (CETREN)

- Italia:
  - Rete Ferroviaria Italiana (RFI)
  - RINA Services S.p.A.
  - Università degli Studi di Roma La Sapienza
- Bélgica:
  - Multitel Asbl
  - Belgorail
- República Checa
  - OLTIS Group a.s.

La propuesta de este proyecto recoge la metodología a llevar a cabo, distribuyendo el trabajo en diferentes paquetes de trabajo, *Work Packages* (WP), cada uno encargado de unas partes del proyecto, como se presenta en la Figura 1.

- **WP0: COORDINACIÓN DEL PROYECTO.** Cuyo objetivo es administrar el consorcio e implantar los mecanismos necesarios para completar el proyecto a tiempo, haciendo el mejor uso posible de los recursos disponibles para alcanzar los objetivos con la calidad deseada.
- **WP1: ESTUDIO DEL ENTORNO DE PRUEBAS.** El objetivo de este paquete es analizar el proceso de trabajo que permita transferir el mayor número de pruebas de vía al laboratorio, optimizando el modelo de pruebas y el intercambio de información.
- **WP2: ARQUITECTURA DE LABORATORIO.** Definición e implementación de arquitectura de laboratorio específica junto con colaboradores de S2R JU.
- **WP3: DEMOSTRACIÓN.** Definición e implementación de pruebas que demuestren la eficacia de la arquitectura propuesta en WP2, así como verificación de la repetitividad de los resultados obtenidos en diversos laboratorios.
- **WP4: VALORACIÓN DEL ENTORNO DE PRUEBAS Y DE LA METODOLOGÍA PROPUESTA.** Valoración del entorno de pruebas propuesto en WP1 y la arquitectura propuesta en WP2 en función de los resultados de las demostraciones llevadas a cabo en WP3.

- **WP5: DIVULGACIÓN.** En este paquete de trabajo se trata de difundir los resultados a toda la Comunidad Ferroviaria, además de asegurar la coordinación con diferentes actividades relacionadas.

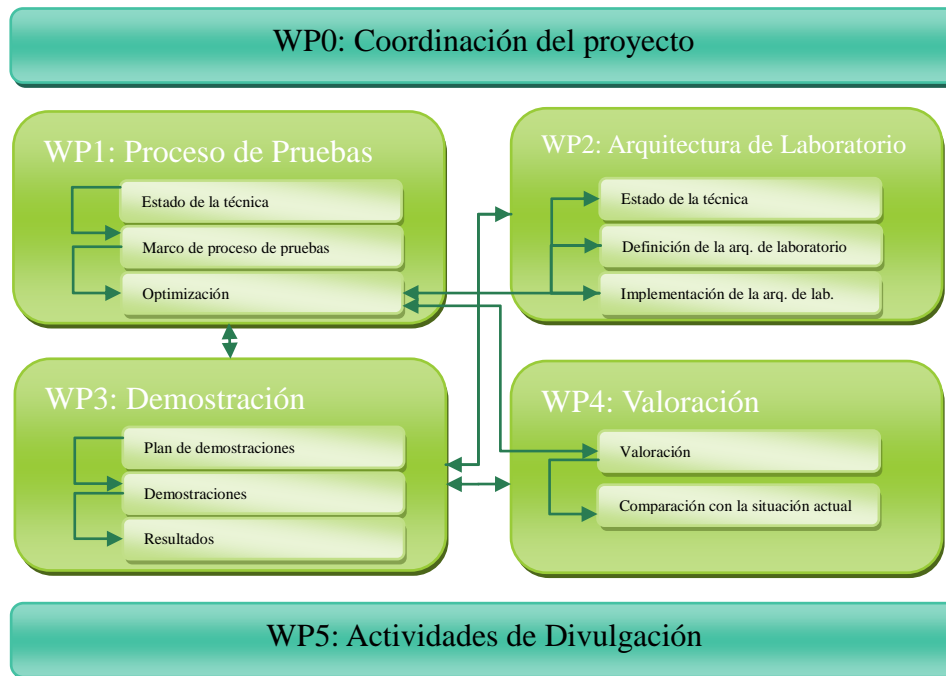


Figura 1. Interdependencia de WPs

## 2.3 OBJETIVO DEL TRABAJO DE FIN DE GRADO: ENSAYOS DE INTEROPERABILIDAD

Dentro del WP3, que trata de demostrar que las pruebas que se llevan a cabo en laboratorio son viables y útiles, existen diversas tareas que deben ser culminadas en unos entregables antes de diciembre del 2017. Una de las tareas dentro de este paquete de trabajo, es que *los datos con los que se trabajará en el entorno de laboratorio, tengan un formato común*, para evitar faltas de entendimiento y compatibilidad entre diferentes países y empresas, puesto que cada una implementa actualmente protocolos diferentes e incluso propietarios.

Parte de los resultados a entregar reúnen ciertos datos sobre la vía en la que se simulan las diferentes pruebas de señalización. Estos datos contienen información acerca de los

puntos y elementos de interés de los que dispone la vía para las pruebas. Esta información proporcionada sobre la infraestructura puede tener que ver con su trazado, es decir por donde pasan las vías, donde se localizan los desvíos, donde finalizan, cuáles son sus longitudes, etc. También pueden incluir información sobre señalización ERTMS u otros tipos de señalización, con elementos o particularidades que se encuentran en la vía como pueden ser estaciones, túneles, restricciones de velocidad. También puede describir conceptos sobre el tipo de material rodante (es decir, con los trenes) que circula por dichas vías, con las rutas que éstos pueden seguir, el tipo de tren, la velocidad máxima permitida para este y un largo etcétera de posibles datos de utilidad para estos ensayos.

En España, el único laboratorio acreditado para realizar las pruebas de interoperabilidad de ERTMS, es el CEDEX, que como se ha comentado previamente, se trata del Centro de Experimentación de obras Públicas adscrito al Ministerio de Fomento. Proporciona apoyo tecnológico en los campos relacionados con la ingeniería civil, edificación y medio ambiente tanto a empresas privadas como a públicas. Este centro reúne varios laboratorios dedicados a pruebas de simulación de ERTMS [5].

Para los ensayos que se realizan actualmente en CEDEX, se solicita a las empresas que faciliten los datos necesarios para la realización de los mismos. Algunas de estas empresas facilitan los datos en formatos propietarios que deben ser traducidos a un estándar para la verificación de interfaces con otros equipos. En este caso, CEDEX utiliza el estándar UNISIG (Subset-112). El problema es que la evolución técnica y las propias empresas del sector, a nivel europeo, parece que se orientan por una plataforma más abierta que ha surgido con fuerza en los últimos tiempos: RailML.

*Puesto que en el ámbito del proyecto VITE se ha decidido emplear el formato de datos que está cobrando más fuerza entre los tecnólogos del ferrocarril (RailML), se hace necesario elaborar un estudio sobre los dos formatos estándar como base para el posterior desarrollo del proyecto VITE.*

*Es decir, el objetivo del presente **Trabajo de Fin de Grado** es realizar una valoración de ambos formatos, mediante el análisis y comparación entre ellos para establecer una eficaz traducción entre ellos, consiguiendo una mayor interoperabilidad de datos con el resto de entidades participantes en el VITE.*

### 3 OBJETIVOS DEL TRABAJO DE FIN DE GRADO

---

Para llevar a cabo el objetivo general del proyecto, deben alcanzarse los siguientes objetivos específicos:

- Realización de un *estudio previo* de toda la documentación relacionada con esta parte del proyecto VITE. Para ello se estudiará la situación de este proyecto en el entorno ferroviario actual, así como el estado del arte de la señalización ERTMS/ETCS y sus ensayos de interoperabilidad.
- *Comparación de ambos formatos* utilizados, UNISIG Subset-112 y RailML, tratando de buscar similitudes y diferencias para así garantizar una posible traducción entre ambos.
- *Análisis* de qué partes del formato empleado en España actualmente (UNISIG Subset 112) son realmente útiles, qué partes son de necesaria inclusión a la hora de la traducción, así como cuáles se podrían obviar por falta de utilidad o de uso.
- Buscar a las partes de necesaria *traducción* su correspondiente en RailML, intentando cubrir toda la información necesaria.
- *Elaboración de propuestas* para la inclusión en RailML de información de utilidad para los usuarios de UNISIG Subset-112.



## 4 ESTRUCTURA DEL DOCUMENTO

---

### **CAPÍTULO 5: ESTADO DEL ARTE**

En este capítulo se introducirán los ensayos de interoperabilidad y el estado actual de la técnica aplicada a dichos ensayos.

### **CAPÍTULO 6: ESPECIFICACIONES TÉCNICAS**

En este capítulo se introducen las especificaciones de los ensayos de interoperabilidad. Se presenta la documentación empleada de los protocolos UNISIG Subset-112 y RailML

### **CAPÍTULO 7: ANÁLISIS COMPARATIVO**

En este capítulo se desarrolla toda la aportación del alumno al proyecto, realizando el análisis comparativo de ambos formatos de ensayos IOP.

### **CAPÍTULO 8: PRESUPUESTO DEL PROYECTO**

En este apartado se presenta el presupuesto aproximado del trabajo

### **CAPÍTULO 9: CONCLUSIONES Y PROPUESTAS**

Se presentan todas las conclusiones sobre el trabajo realizado y propuestas de mejora al proyecto en el que este TFG se encuadra

### **CAPÍTULO 10: GLOSARIO DE TÉRMINOS**

### **CAPÍTULO 11: REFERENCIAS BIBLIOGRÁFICAS**

Se incluyen en este apartado todas las fuentes bibliográficas empleadas

## 5 ESTADO DEL ARTE

---

Desde el nacimiento del sistema ferroviario, se ha tenido la necesidad de controlar la operación de trenes para evitar colisiones y optimizar el número de circulaciones posibles. Esto se ha hecho siempre tratando de separarlos, intentando que en la zona en la que se sitúa un tren, denominada cantón, no pueda entrar ningún otro.

Para conseguir esto siempre se ha tratado de desarrollar sistemas de gestión de tráfico ferroviario, desde un comienzo con operarios informando de la ocupación de la línea, pasando por señales luminosas y señales fijas, hasta llegar a sistemas con tecnología más moderna, introduciendo sistemas automatizados denominados ATP (Automatic Train Protection). Estos sistemas de señalización controlan la velocidad del tren, adaptándola a la velocidad óptima que le permita mantenerse en condiciones seguras [6].

Todos estos sistemas ATP proveen las señales correspondientes al tren de diferentes modos [2]. Esto se puede hacer de manera puntual a través de transpondedores denominados balizas situadas en la vía que proporcionan al tren la información necesaria en puntos específicos. Este es el caso del sistema de señalización española ASFA.

También se puede realizar de manera continua, a través de lazos de cable situados en la vía de gran longitud, que transmiten al tren la información necesaria. Este es el caso del sistema alemán también empleado en España (en la línea Madrid-Sevilla): LZB.

Es también posible transmitir todos estos datos de señalización de manera continua y a distancia a través de señales de radio (ERTMS: ETCS + GSM-R).

En la actualidad, la red ferroviaria europea cuenta con diferentes sistemas de señalización dependiendo de la línea y el estado donde se encuentre, como muestra la Figura 2. Esto genera dificultades en la circulación, especialmente en las fronteras, donde provoca interrupciones e incluso imposibilidad de circulación a través de ellas debido a la falta de compatibilidad entre los diferentes sistemas. Esto da como resultado que en los trenes que deban traspasar fronteras entre países con diferentes señalizaciones deben ir equipados con ambos sistemas y los maquinistas deben saber cómo utilizar ambos. Estas dificultades provocan a su vez un obstáculo para la seguridad, ya que el sistema de

Map of Europe showing the distribution of various rail vehicle types by country. The map is color-coded by country and labeled with vehicle codes. Labels include: GB ATP, RETB, TPWS, TVM 430; ATB First generation, ATB new generation; ZUB 123; ZUB 121; INDUS/PZB, LZB; EVM; BACC, RSDD/SCMT; ASFA; LZB, EBICAB; EBICAB 700; ATP-VR/RHK; LS; Crocodile; KVB, TVM 300, TVM 430; and Crocodile.

La mejora de la interoperabilidad y de las tecnologías de señalización aportaría a la industria ferroviaria la capacidad de mejorar y optimizar el tráfico tanto de viajeros como mercancías. De esta manera se podrían reducir las distancias entre trenes operativos y así mejoraría la industria ferroviaria su competitividad y su posición en el mercado mundial, suponiendo de este modo una alternativa viable a la circulación por carretera. Por otra parte, la estandarización de la señalización simplificaría la formación de los maquinistas minimizando los posibles errores humanos, además de implicar una reducción significativa de costes, ya que solo habría necesidad de la fabricación de una clase de equipo ATP. De este modo, los trenes que realizasen trayectos internacionales solo deberían llevar un equipo de señalización embarcado, lo que también llevaría a una reducción de costes, ya que actualmente llevan hasta siete equipos diferentes. Todo ello redundaría en una mejora en seguridad y un tráfico fluido e ininterrumpido en las fronteras.



## 5.1 ERTMS

El sistema de señalización ERTMS es aquél que ha adoptado la UE para salvar dichas diferencias entre los diversos sistemas de señalización. Comenzó a consolidarse a partir de 2005, aunque se llevaba desarrollando desde 1989, y actualmente es el sistema de señalización estándar europeo según la Directiva 96/48/CE sobre la interoperabilidad de los Sistemas Ferroviarios Transeuropeos de Alta Velocidad[5].

Los sistemas ERTMS=ETCS+GSM-R constan de diferentes componentes[2]:

- *European Train Control System (ETCS)*

El sistema europeo de control de trenes está a cargo de los sistemas de seguridad como el control automático de trenes (ATC) y de la protección automática de trenes (ATP), que son los sistemas encargados de leer aquella información transmitida desde la vía para activar el frenado de emergencia en el caso de que el tren circule sin cumplir algunos requisitos impuestos para garantizar la seguridad. De este modo, los sistemas ETCS son la base de la tecnología ERTMS.

- *Global System Mobile – Railway (GSM-R)*

Es el sistema de transmisión por telefonía móvil GSM adaptado a la frecuencia ferroviaria. Se emplea para enviar voz y datos necesarios para ETCS.

- *European Traffic Management System (ETMS)*

Es el sistema europeo de gestión de tráfico, a cargo de la gestión de los recursos para cumplir con las necesidades del tráfico ferroviario como administrar los horarios, la capacidad de cada línea, la flota de trenes, etc.

Los sistemas ERTMS tienen varios niveles de aplicación que se adaptan a las necesidades de cada proyecto, debiendo un sistema embarcado poder circular con un Nivel X y todos los anteriores. Los niveles de aplicación de ERTMS/ETCS son los que siguen[2]:

- *Nivel 0*

Corresponde con el caso en el que un tren equipado con un equipo ERTMS circula por una línea no equipada, o equipada con un sistema que no es compatible o que no se encuentra operativo. Para la operación de este nivel se emplea señalización externa, como señales o carteles.

- *Nivel NTC (National Train Control System)*

En líneas con este nivel, el tren dispone de sistemas de a bordo ERTMS/ETCS, pero circula por una vía con un sistema de señalización nacional. De este modo, el tren debe llevar a bordo, a parte del sistema de lectura ERTMS/ETCS, un sistema NTC para la lectura e interpretación del sistema nacional correspondiente.

- *Nivel 1*

El tren dispone de dispositivos ERTMS/ETCS a bordo. Se produce una supervisión continua de la velocidad del tren pero con transmisión puntual a través de Eurobalizas, es decir, balizas del sistema ERTMS/ETCS. Para solventar los problemas que supone la transmisión no continua de información debido al uso de balizas, en este nivel se añade en ocasiones el uso de Eurolazo (cable radiante) o sistemas de radio in-fill.

- *Nivel 2*

En este nivel se efectúa una supervisión continua de la velocidad, así como una transmisión continua. Para esto se emplean, además un sistema de señalización subyacente con Eurobalizas como puntos de referencia de posicionamiento, señales de radio mediante el sistema de Euroradio. Con estas señales de radio se transmiten las autorizaciones de movimiento, la información sobre la vía, etc.

En este nivel es necesario el uso de Centros de Bloqueo por Radio (RBC) que son los encargados de la comunicación bidireccional con cada uno de los trenes. Todo este sistema hace opcional el uso de señales laterales.

- *Nivel 3*

En este nivel no se requiere un sistema de señalización subyacente, sino que toda la transmisión de información y la supervisión se realizan por radio con el RBC.

El sistema ERTMS ya está instalando en muchos países Europeos, entre los que se encuentra España, que cuenta con esta tecnología en unos 1850 km de línea, entre ellas todas sus líneas de Alta Velocidad. De este modo España es el país con mayor implantación de este sistema en el mundo[8].

Además de en Europa, esta tecnología se está abriendo paso poco a poco al resto del mundo, existiendo ya algunas líneas y numerosos proyectos en países como China y Arabia Saudí, como muestra la Figura 3.

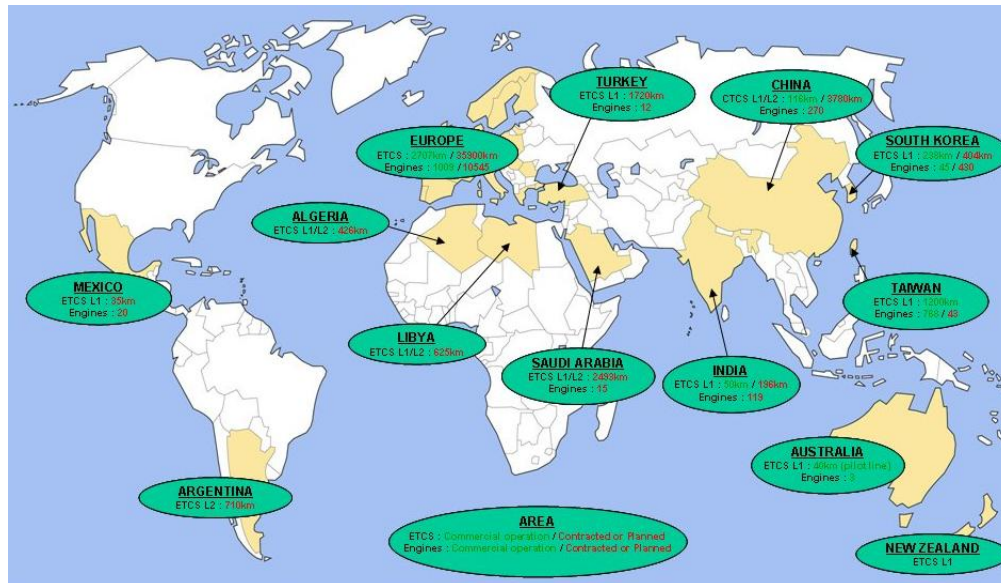


Figura 3: Implementación de ERTMS a nivel mundial[9]

## 5.2 ENSAYOS DE INTEROPERABILIDAD ERTMS/ETCS

A pesar de estar ya presente en muchos países, el proceso de implementación de ERTMS en el sistema ferroviario europeo está siendo un proceso muy lento y costoso, ya que requiere para su implantación fondos públicos, así como coordinar a equipos de trabajo de diferentes países.

Además, el hecho de poner en funcionamiento una tecnología aún en desarrollo supone realizar un gran número de ensayos y pruebas, lo que conlleva grandes inversiones de tiempo y dinero.

Entre estos ensayos destacan los ensayos de interoperabilidad (IOP), que son aquellos destinados a comprobar que equipos de diferentes fabricantes pueden funcionar trabajando juntos.

Por ejemplo, consiste en comprobar si un equipo embarcado fabricado por el fabricante A se puede comunicar sin problemas con balizas del fabricante B.

Hasta ahora la mayor parte de los ensayos IOP se realizan in situ, es decir, en la infraestructura real. Esto supone grandes inversiones de tiempo y dinero con un amplio

despliegue de recursos. Por ello, es muy importante reducir costes y tiempos en este tipo de pruebas.

Con este objetivo nace el subproyecto VITE, parte del proyecto S2R, mencionados en el apartado 2.2. Con este proyecto se trata de llevar todas estas pruebas a laboratorio, probando su fiabilidad, aplicabilidad y beneficios.

Es un proyecto aún en fase de desarrollo, sin todavía resultados concluyentes. No obstante, a pesar de que dentro del proyecto no se haya llegado a ningún resultado, ya hay pruebas realizadas en laboratorios.

En España, el CEDEX cuenta con unas instalaciones denominadas Laboratorio de Interoperabilidad Ferroviaria (LIF) en el que ya se están realizando ensayos de este tipo (Figura 4). El LIF está a cargo de prestar apoyo a la implantación de líneas con señalización ERTMS de niveles 1 y 2, así como de la estandarización de especificaciones técnicas con la participación en proyectos como el VITE. De este modo se realizan aquí pruebas normalizadas de Eurobaliza y equipos embarcados, así como ensayos de interoperabilidad simulados y sobre líneas reales[5].



*Figura 4: Laboratorio de Interoperabilidad Ferroviaria [5]*

Partiendo de todos los ensayos ya realizados y de todas las especificaciones ya disponibles sobre sistemas ERTMS, el proyecto VITE continuará mejorando y estandarizando todas las normativas necesarias para lograr implantar esta señalización en toda Europa e intentar seguir exportando esta tecnología a países del resto del mundo.

## 6 ESPECIFICACIONES TÉCNICAS

---

Para lograr la operación segura en la red ferroviaria, es imprescindible el intercambio eficaz de grandes cantidades de información entre diferentes dispositivos que forman todo el sistema ERTMS. Específicamente, resulta especialmente importante la comunicación efectiva entre los dispositivos en vía y aquellos embarcados.

Las buenas comunicaciones aseguran las distancias seguras entre trenes, sin ser excesivas, ya que resultaría en una operación poco eficiente. Por lo tanto, una tarea importante a la hora de garantizar la seguridad, es asegurar que ese intercambio de información esencial se produce de manera correcta.

Para que estas comunicaciones resulten efectivas, es necesario que los datos estén en el mismo "idioma". Es decir que la literatura de esos datos esté escrita y estructurada bajo el mismo formato y siguiendo las mismas especificaciones.

Esto genera un gran problema de interoperabilidad, ya que dos dispositivos programados para intercambiar información en diferentes formatos no se pueden comunicar de forma eficaz, ya que la información enviada se vería malinterpretada.

Además, la falta de un formato común lleva a problemas en la conversión y a malas interpretaciones de datos, lo que desemboca inevitablemente en costes extra que se podrían suprimir si no se tuviese este problema.

Es por esto que diversos investigadores y desarrolladores, están trabajando con formatos que estén altamente disponibles, evitando así estos problemas.

El proyecto INESS (Integrated European Signaling System) supone un buen respaldo para el proyecto VITE en cuanto a la elección del formato con el que se va a trabajar.

INESS hizo un estudio valorando seis formatos de datos diferentes, siendo uno de los requisitos a valorar la compatibilidad con ERTMS. Los formatos valorados son[10]:

- RailML
- UNISIG (Subset-112)
- EuroIXL
- Stamp

- DB-Model
- PoE Siemens

Con las valoraciones de INESS, el proyecto VITE decidió centrarse en **RailML** y **UNISIG (Subset-112)**, por ser los más recomendados por dichos resultados[11].

Ambos formatos se estructuran estableciendo su base en archivos XML.

## 6.1 XML SCHEMA DEFINITION

El lenguaje XML (Extensible Markup Language) es un lenguaje basado en etiquetas desarrollado por W3C (World Wide Web Consortium). Se trata de un lenguaje meramente descriptivo, es decir, no realiza ninguna acción, sino que solo sirve para el intercambio de información. Una gran ventaja que ofrece este lenguaje es que puede ser leído tanto por un usuario como por una máquina, por lo que resulta de gran utilidad a la hora de definir estructuras de datos[12].

Todos los datos aquí encuadrados están recogidos entre etiquetas de apertura o cierre como puede ser el ejemplo siguiente:

```
<pruebasERTMS>
    <formato> UNISIG Subset 112 </formato>
    <formato> UNISIG Subset 112 </formato>
</pruebasERTMS >
```

De este modo se describen los dos formatos dentro de la etiqueta `pruebasERTMS`.

El formato de archivos que utilizan RailML y UNISIG (Subset-112) es un derivado de XML denominado XSD (XML Schema Definition). Los esquemas de XML están formados por elementos y atributos, todos ellos anidados formando una estructura de datos compleja.

Es así como cada *elemento* está formado por varios elementos hijos, que pueden ser de opcional u obligada inclusión, que además tienen límites que indican la cantidad mínima y máxima de un elemento que su elemento continente puede tener.

Por otra parte, cada elemento tiene *atributos*, que describen sus cualidades, teniendo cada uno un tipo de dato. Hay ciertos tipos de datos básicos que están predefinidos en XSD para números, cadenas de caracteres, fechas, etc.

XSD incluye también la posibilidad de incluir anotaciones y comentarios a sus elementos, de este modo posibilitando añadir más documentación o enlaces donde encontrarla.

Con todos estos datos se pueden crear esquemas gráficos mostrando de una forma intuitiva la estructura correspondiente. En la Figura 5 se muestra el esquema de los componentes principales de RailML[13].



Figura 5: Esquema *railml* [13]

Se puede observar de una forma clara e intuitiva que el elemento *railml* consta de una secuencia ( ) de cinco elementos hijos opcionales (indicados por línea discontinua), cada uno con una estructura compleja subyacente (desplegable con el símbolo ). Además, cada elemento incluye una descripción.

De este modo, mediante lenguaje de etiquetas se puede ir definiendo el esquema correspondiente a la estructura de datos que se desea representar.

Para trabajar más fácilmente con los documentos XSD publicados por RailML, se ha empleado la herramienta de software Altova XML Spy, que permite una fácil creación, modificación y visualización de esquemas XML.

## 6.2 FORMATOS DE DATOS ERTMS

A continuación, se describen los dos formatos en los que se enfoca el proyecto VITE, con sus características, fortalezas y debilidades.

### 6.2.1 UNISIG (Subset-112)

El grupo UNISIG (Figura 6) es un consorcio de industrias ferroviarias fundado en 1998/99 bajo petición de la CE, con el objetivo de crear las especificaciones de los requerimientos de los sistemas ERTMS[14].



Figura 6: Logo de UNISIG

Está encargado, no sólo de desarrollar dichas especificaciones, sino de mantenerlas y renovarlas según sea necesario.

Dentro de UNISIG, el grupo de trabajo IOP UNISIG Working Group es el encargado de crear las especificaciones para ensayos IOP, tratando de llegar a un conjunto de normas que se puedan generalizar en el uso de tecnología ERTMS, ya que la CE se ha comprometido a financiar estudios futuros solo si se han desarrollado unas especificaciones estándar [15].

Este grupo ha desarrollado un conjunto de documentos que recogen las especificaciones de los requisitos técnicos que tienen estas pruebas. Estos documentos están clasificados en Subsets (SS), cada uno de los cuales trata un punto diferente.

Entre todos estos documentos, son de especial interés para este proyecto aquellos que tratan la información intercambiada entre dispositivos ERTMS en los ensayos IOP.

- **SS-110:** *UNISIG Interoperability Test - Guidelines (Ensayos de interoperabilidad - Reglas generales)*

Este SS supone una introducción a los ensayos de interoperabilidad, incluyendo una descripción general de los mismos.

- **SS-026:** *System Requirements Specification (Especificaciones de los requerimientos del sistema)*



En este documento recoge las especificaciones de los mensajes enviados entre dispositivos ERTMS. De importancia para este proyecto son los capítulos 7 y 8.

En el capítulo 7 quedan reflejadas todas las variables intercambiadas en los mensajes, explicando su significado e indicando todos los valores posibles que dicha variable puede tomar, con sus respectivas correspondencias con cada significado.

Por su parte, el capítulo 8 incluye cómo se deben transmitir estos mensajes, es decir, en qué orden, cuánta memoria es preciso reservar para cada mensaje, etc.

- **SS-054:** *Responsibilities and rules for the assignment of values to ETCS variables (Responsabilidades y reglas para la asignación de valores a las variables ETCS)*

En este SS se recogen las diferentes normas que aplican a variables del SS-026, incluyendo modificaciones, eliminaciones, y alguna especificación extra.

- **SS-112:** *UNISIG Basics for Interoperability Test Scenario Specifications (Bases para las especificaciones del escenario de pruebas de interoperabilidad)*

En este documento se incluye la especificación de un marco generalizado de pruebas de interoperabilidad, definiendo todos los datos de vía intercambiados. Aquellos datos que tengan valores que precisen explicación se encuentran definidos en el SS-026.

Este es el documento principal de UNISIG en el que se basa este proyecto, del que se va a utilizar la versión última, la 3.6.0.

Al margen de estos documentos, UNISIG también se basa en algunos documentos redactados por la Agencia Ferroviaria Europea, como las *Especificaciones Técnicas de Interoperabilidad para la Infraestructura*.

Aunque el SS-112 indica que el formato común aún no está desarrollado, está comprobado que este documento ofrece una estructura bastante completa para un esquema básico de la información necesaria para un sistema ERTMS de Nivel 1 y Nivel 2.

Sin embargo, este sistema tiene algunas carencias para la definición de algunos elementos, y algunas otros datos pueden dar lugar a malinterpretaciones.

Por otro lado, existe el problema que plantea el hecho de que estas especificaciones no sean de uso abierto, y existan dificultades para conseguir los derechos de uso.

Todo esto hace que, a pesar de que este formato tenga un alto potencial, ya que su nivel de definición es mayor que el de ningún otro, este formato no sea ampliamente empleado. Su uso dentro de empresas no está documentado, pero es el formato utilizado principalmente en laboratorios españoles como el LIF del CEDEX[11].

### 6.2.2 RailML

RailML (Railway Markup Language) (Figura 7) es un lenguaje descriptivo de código abierto basado en esquemas XML, destinado al intercambio de datos en el entorno ferroviario. La creación de este lenguaje comenzó en 2002



*Figura 7: Logo de RailML*

por un grupo de accionistas ferroviarios en el afán de estandarizar el intercambio de datos en la industria ferroviaria, así como de simplificarlos, con el objetivo de facilitar una sencilla interconexión entre diferentes aplicaciones ferroviarias[13].

El desarrollo de RailML es un proceso de colaboración entre diferentes empresas ferroviarias, de software y consultorías que va evolucionando según se identifican más necesidades del sistema.

De hecho, cualquier empresa está invitada a colaborar en el proyecto, y toda su documentación aprobada y sus esquemas están disponibles online en su página web oficial, ofreciendo así un desarrollo y mantenimiento completamente transparente.

Además, disponen de una página wiki relacionada, en la que se explica con detalle toda la documentación de la última versión de RailML, la 2.3, publicada en 2006. En esta página wiki se invita a personas relacionadas con el proyecto a crearse una cuenta y colaborar a la mejora de dicha página, dando siempre prioridad a las publicaciones de los colaboradores experimentados [16].

En este formato se incluye información acerca de la infraestructura, el material rodante (trenes), las rutas y conexiones, y los horarios de operación, ofreciendo así una estructura completa acerca de la información necesaria para cualquier aplicación ferroviaria.

Aun así, la mayoría de elementos de RailML incluyen el elemento y el atributo `other`, el cual permite añadir la información adicional que necesite cualquier desarrollador en la creación de esquemas específicos para otras aplicaciones ferroviarias. De este modo, se ofrece un formato con una gran flexibilidad y con grandes posibilidades de convertirse en el formato estándar.

Hasta la fecha la versión última desarrollada es la 2.3, aunque se está trabajando actualmente en la 3.0 con la colaboración de otro proyecto denominado RailTopoModel.

Toda la documentación de este proyecto se encuentra escrita en inglés, para facilitar la extensión de su uso. Además parte de la documentación se encuentra traducida al alemán y al francés.

Por otra parte, se ha probado que este formato satisface los requisitos del proyecto INESS, que supone de gran importancia para el proyecto VITE. En su clasificación de aptitud, RailML quedó segundo en el ranking general y primero para las aplicaciones ERTMS, que es el campo de interés fundamental del VITE.

Además, el uso de RailML es cada vez más extenso, no solo en Europa sino también en el resto del mundo. La mayoría de empresas de peso del sector de la administración ferroviaria son colaboradores actualmente de RailML, entre las que se incluyen empresas como ADIF, Infrabel, ÖBB, ProRail, etc.

Por todo esto, *se ha considerado RailML como el formato más indicado para definir un estándar, y por lo tanto, se ha escogido finalmente como el formato por excelencia para el proyecto VITE.*

## 7 ANÁLISIS COMPARATIVO

---

La definición de todas las características de la vía se puede hacer de muchos modos diversos, estructurando la información de manera muy diferente. Esto sucede con los formatos ofrecidos tanto por el protocolo empleado en España como en el estándar europeo, dando lugar a grandes impedimentos de interoperabilidad. Es por esto por lo que resulta fundamental hacer una traducción de un formato a otro, para hacer la red ferroviaria española más competitiva de fronteras para afuera.

Tal y como se ha comentado previamente, existen dos protocolos utilizados:

- **UNISIG (SS-112)**, es el que actualmente se emplea en España, desarrollado por empresas del sector ferroviario.
- **RailML**, se está imponiendo su uso como estándar en Europa por parte de los usuarios, que a su vez contribuyen a su desarrollo.

### 7.1 UNISIG (SUBSET-112)

Por una parte, en el protocolo Subset-112 publicado por UNISIG, cuyo uso está extendido en el entorno de pruebas nacional, la estructura de la información acerca de la vía se encuentra toda dentro del elemento `TrackSide`, que representa el grueso de la definición de todos los elementos de la infraestructura en tierra necesarios para el desarrollo del escenario de pruebas ERTMS, englobando la información necesaria de todos los puntos y elementos clave de la vía [17].

Este elemento `TrackSide`, es del tipo de datos con mismo nombre, y supone un tipo complejo de datos que encuadra cuatro listas de diferentes tipos de elementos:

#### 1) *Elementos topológicos.*

Los elementos topológicos son aquellos que definen la "forma" de la red, es decir, una especie de mapa físico simplificado, definiendo así un esquema de cómo es el trazado de la red, estableciendo donde hay desvíos, cruces, cambios kilométricos, donde termina la vía, etc.

2) *Elementos de la vía*

Los elementos de vía suponen otro tipo de información asociada a un punto o un área de la topología antes descrita, indicando así la localización y características de señales, cambios de velocidad, cambios en diversos parámetros, etc.

3) *Rutas*

Las rutas recogen partes de la topología que suponen itinerarios que los trenes pueden seguir,

4) *Propiedades de la vía*

Las propiedades de vía recogen datos acerca de los proveedores y administradores de la infraestructura

Además, en estos elementos viene implícito un sistema de posicionamiento para localizar cada uno de los elementos y características en puntos concretos de la infraestructura, así como el método de visualización de los mismos para crear una interfaz sencilla de utilizar en los ensayos de interoperabilidad.

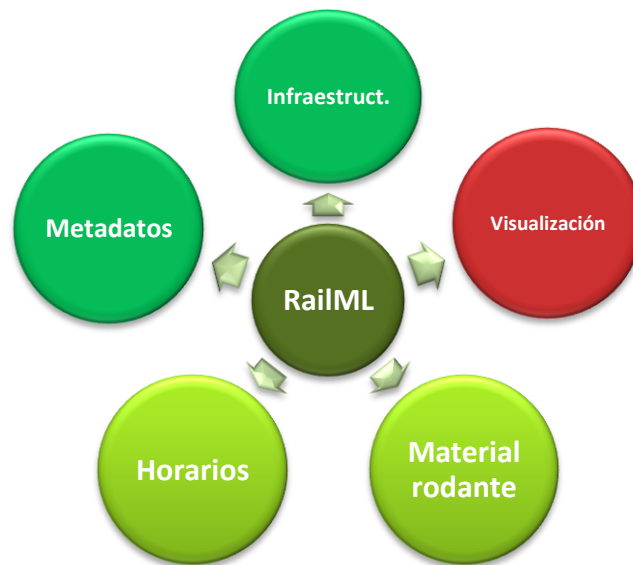
En la Tabla 1 podemos observar un primer acercamiento a la estructura de datos provista por UNISIG con respecto a los datos de la infraestructura.

Desglose de la infraestructura ferroviaria			
Elementos topológicos	Elementos de la vía	Rutas	Propiedades de la vía (opcional)
Segmento	Señal	Ruta	Información de la Administración
Punto	Baliza		Nombre de la vía
Conexión	Eurolazo		Información de la versión de vía
Final	RIU		Fecha
Conexión	Ocupación de vía		Proveedor de RBC
Externa	Cambio de ancho de vía		Versión de RBC
	Cambio de voltaje		Proveedor de vía
	Cambio de carga por eje		Versión de vía
	Área de cambio de sentido		
	Área de desviación		
	Cambio de valor nacional		
	Cambio de velocidad estática		
	Cambio de gradiente		
	Condición de la vía		

Tabla 1: Estructura del desglose en tierra

## 7.2 RAILML

Por su lado, RailML, que supone el formato a establecer como estándar en el resto de Europa, tiene una estructura que engloba muchos más campos, ya que además de todos los datos sobre la infraestructura que incluye UNISIG en su especificación, también aporta muchos más datos sobre la misma, así como información detallada sobre el material rodante, es decir sobre los trenes que van a circular por esa infraestructura, y acerca de horarios y temporización de rutas, paradas, esperas, etc. De este modo, intenta englobar todos los datos fundamentales recogidos por las pruebas de diversas empresas ferroviarias.



*Figura 8: Estructura básica RailML.*

Dentro de esta estructura, representada en la Figura 8 vamos a emplear únicamente el elemento `infrastructure`, y algún dato general de metadata (metadatos), que son los campos que recogen la información sobre la infraestructura y la administración respectivamente, quedando `rollingstock` material rodante) y `timetable` (horario) vacíos, ya que no están cubiertos en las especificaciones de UNISIG.

Por otra parte, el elemento `infrastructureVisualizations`, que define la visualización de los datos recogidos, sí que resultaría muy útil para modelar la visualización de la infraestructura durante las pruebas. Sin embargo, este elemento se ha dejado de emplear desde la versión anterior de RailML, la 2.1, por lo que en la 2.3 no se empleará.

## 7.3 ELEMENTOS TOPOLÓGICOS

La topología de vía hace referencia a la manera de representar las vías y líneas que componen la infraestructura ferroviaria, generando así un mapa en modo de esquema, recogiendo únicamente la información necesaria. En ambos formatos analizados se emplea un sistema de "líneas" que representan tramos de la red, unidos entre ellos dando forma a la infraestructura completa. Los tipos de conexiones entre diferentes tramos representa diversas estructuras que se pueden encontrar en una vía como pueden ser desvíos, cruces, etc.

### 7.3.1 Topología de UNISIG (Subset-112)

Por un lado, en el Subset-112 de UNISIG se emplea para la definición de la topología una estructura de segmentos y puntos de unión, unidos entre sí mediante nodos de conexión como se puede ver en el esquema de la Figura 9.

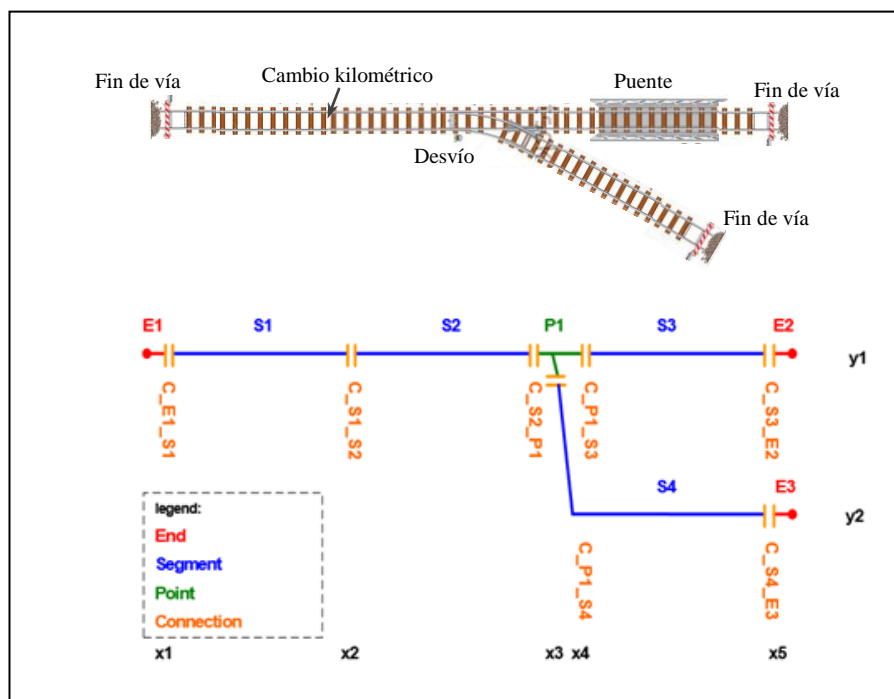


Figura 9: Ejemplo de topología en Subset 112



Toda la información acerca de la estructura topológica se encuentra englobada bajo el elemento `topologyElements`, del tipo complejo `topologyElementList`, que engloba cinco tipos de elementos:

- **segments:** vector de elementos del tipo `segment`, que definen un tramo de línea delimitada por dos conexiones (`connections`). En la Figura 9 se muestran en azul.

Los segmentos de UNISIG representan tramos de línea en los que no existe ninguna conexión con ningún otro segmento, en los que no existe ningún cambio kilométrico, ningún fin de línea, descarrilador, ni ningún otro tipo de variación topológica. Están definidos por cuatro atributos:

- Un identificador (`id`) de tipo `string` que ha de ser único.
- Un vector de elementos del tipo `bend` (curva) que cuenta con el punto kilométrico donde se halla y las coordenadas `x`, `y`. Este atributo referente a las coordenadas tiene únicamente fines de visualización, las coordenadas son en 2D y no son reales, sino puntos en el plano para poder dibujar las curvas en una visualización de la línea.
- Dos conexiones referenciadas con sus `id`, para representar el inicio y final del segmento
- **points:** vector de elementos de tipo `point`, que definen partes adimensionales con tres conexiones, usadas para unir tres segmentos. En la Figura 9 se muestran en verde. Los puntos cuentan con:
  - Un identificador (`id`) de tipo `string` que ha de ser único.
  - Tres conexiones referenciadas con su `id`, para representar la unión los segmentos que une. La primera conexión corresponde a aquella en la dirección de cola, la segunda aquella que va a la izquierda, y la tercera a la derecha.
- **ends:** vector de elementos del tipo `end`, que marcan punto final adimensional de una línea. En la Figura 9 se muestran en rojo. Los finales de línea tienen los siguientes atributos:
  - Un identificador (`id`) de tipo `string` que ha de ser único.

- Una conexión referenciada con su id, para representar la unión con el resto de la línea.
- **connections:** vector de elementos de tipo `connection`, que representan elementos adimensionales que unen dos elementos de los tipos anteriores. En la Figura 9 se muestran en naranja. Las conexiones, cuentan con los siguientes atributos:
  - Un identificador (`id`) de tipo string que ha de ser único.
  - Unas coordenadas `x`, `y` para representar su visualización.
  - Dos elementos denominados `trackPartId1` y `trackPartId2` de tipo string que hacen referencia al identificador de las respectivas partes que la conexión une (segmentos, finales, puntos).
  - Dos elementos denominados `kp1` y `kp2` del tipo `double`, que corresponden a los puntos kilométricos de los elementos que une la conexión. Pueden tener valores dispares en el caso de que esta conexión represente un cambio kilométrico, que se pueda producir, por ejemplo, en un desvío.
- **externalConnections:** vector de elementos de tipo `externalConnection` que representan la unión de uno de los elementos anteriores (salvo conexiones) con una parte de vía que se encuentra fuera de los límites de la definición de la línea actual, probablemente en una línea definida en otro archivo, y cuentan con los atributos:
  - Un identificador (`id`) de tipo string que ha de ser único.
  - Unas coordenadas `x`, `y` para representar su visualización.
  - Una conexión referenciada con su id, para representar la unión con el resto de la línea.
  - Un elemento denominado `externalTrackside` de tipo string que identifica el nombre de la línea externa a la que está conectada.
  - El identificador (`id`) de la conexión externa con la que está conectada.

En el caso de que se desconozcan los datos sobre la línea externa a la que está unida, se tratará como un final de línea.

### 7.3.2 Topología de RailML

Por su parte, en RailML, no existe un elemento único que defina la topología de la línea, sino que dentro de la definición de la infraestructura completa existe un conjunto de segmentos que componen la red. Dentro de cada segmento, denominados `track` en este protocolo, se encuentra incluido un elemento de obligada inclusión definiendo su topología (`trackTopology`).

Este elemento define la topología de cada segmento incluyendo elementos que definen su comienzo y su fin (`trackBegin` y `trackEnd` respectivamente) y que determinan qué es lo que ocurre en cada uno, es decir, a qué van conectados, o si se trata de un final de línea. Esto se especifica dentro de dichos elementos de fin o comienzo de línea, que incluyen un nodo a elección (solo puede haber uno entre todas las opciones) entre cuatro elementos diferentes, que corresponden con:

- **connection**: representa una conexión con otro `track` de la infraestructura, y posee su propio `id` y una referencia al `id` del segmento al que se conecta (`ref`)
- **bufferStop**: representa un final de línea con un tope amortiguado que para el tren, como el representado en la Figura 10. De sus atributos solo resulta de interés su `id`, los demás no presentan ninguna correspondencia relevante con UNISIG.



Figura 10: Tope de final de vía (buffer stop) en la estación de Barcelona [8]

- **openEnd**: indica que el modelador de la línea no conoce la continuación de la línea, o que la parte a la que está conectada no forma parte del archivo actual. Del mismo modo que en los topes de vía, solo resulta de interés en este estudio su identificador o `id`.

- **macroscopicNode**: sirve para modelar la unión de un conjunto de segmentos sin importar su unión física, como puede darse en una estación, donde una gran cantidad de segmentos pueden verse interconectados, sin estar su unión en el mismo exacto punto geográfico. Debido a sus características, no es empleado en esta comparativa.

Al margen de los extremos de vía existe otro elemento opcional que es `connections`, que representa las conexiones extra en cualquier punto del segmento, a parte de las conexiones principales en los extremos del segmento. Estas conexiones extra, cuya cantidad no está limitada, pueden ser de dos tipos:

- **switch**: elemento que define una aguja, como la que aparece en la Figura 11. Una aguja es un punto de una vía que puede disponer de conexiones hasta con tres vías diferentes, permitiendo mediante una instalación mecánica el paso de los trenes a una vía u otra indistintamente desde la vía principal. Dispone de entre uno y tres elementos de tipo `connection`, que tienen como atributos obligatorios su propio `id`, el atributo `ref` que incluye el `id` del elemento al que está conectado, y la orientación del tráfico que va por esa rama ("incoming" si es entrante y "outgoing" si es saliente)



*Figura 11: Aguja en la estación de Villa Joyosa [8]*

- **crossing**: elemento que define un cruce de vías, lo que supone que dos o más vías se cruzan físicamente pero no se puede realizar el cambio entre ellas, como el mostrado en la Figura 12.



Figura 12: Antiguo cruzamiento en Palencia [8]

A parte de estos elementos hijos, los elementos representativos de comienzo y final de vía, y las conexiones `switch` y `crossing`, también disponen de algunos atributos de interés:

- **id**: Un identificador de tipo `string` que ha de ser único y de obligada inclusión.
- **pos**: indica la posición que ocupa el elemento en relación con al comienzo del segmento. Este atributo es obligatorio y del tipo `tLengthM`, que define con un número decimal una distancia en metros.
- **absPos**: indica la posición que ocupa el elemento con su punto kilométrico. Este atributo es opcional y del tipo `tLengthM`, que define con un número decimal una distancia en metros.

*Teniendo en cuenta las estructuras definidas para ambos formatos se establece en el siguiente punto cómo modelar la topología en RailML a partir de las estructuras ofrecidas en el Subset-112 de UNISIG.*

### 7.3.3 Traducción de la topología de Subset-112 a RailML

Para establecer una correspondencia entre ambos formatos, se irán estudiando cada uno de los elementos o combinaciones de elementos contempladas en el documento Subset-112 y encajándolos en las estructuras que ofrece RailML, realizando así la traducción del formato español al europeo, especificando los ejemplos de código en verde en las figuras.

### 7.3.3.1 Conexión de dos segmentos

Las conexiones de dos segmentos según UNISIG se realizan mediante un tercer elemento `connection` que los une, mientras que en RailML se realizan mediante nodos de conexión en sus extremos.

Los puntos kilométricos que se indican en las conexiones de UNISIG se establecerán en RailML dentro de los atributos de posición de los extremos del segmento.

Este caso se puede ilustrar con un ejemplo en el que se produce un cambio kilométrico entre dos segmentos, mostrado en la Figura 13.

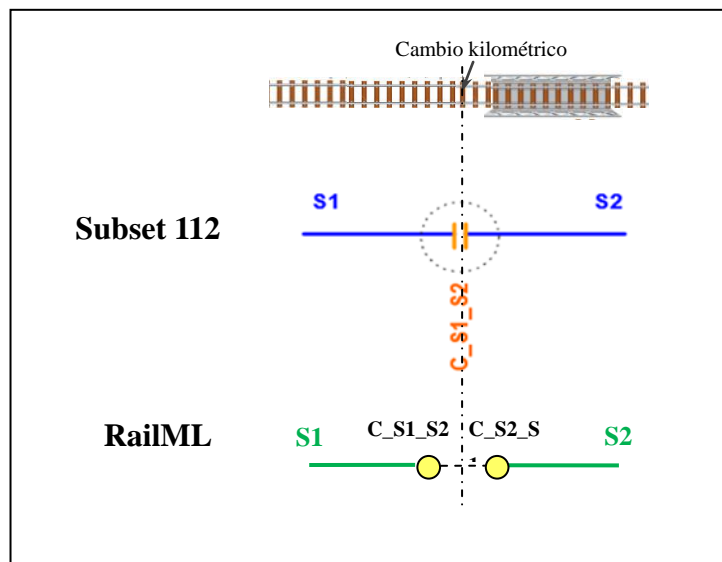


Figura 13: Cambio kilométrico

En el ejemplo de conexión mostrado, según la especificación de UNISIG se definiría como se muestra en la Figura 14, conectando los segmentos S1 y S2 que aparecen referenciados dentro de la conexión C\_S1\_S2. Esta conexión supone un cambio kilométrico en la línea, pasando del PK (punto kilométrico) 300 al final de S1 al PK 400 en el comienzo de S2.

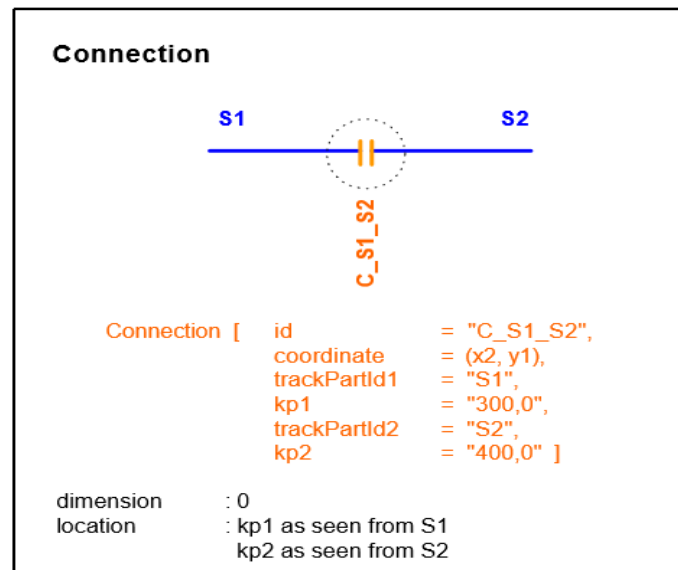


Figura 14: Ejemplo de conexión entre dos segmentos en Subset-112

En RailML, la misma conexión de dos segmentos se modelaría como se especifica en la Figura 15, es decir, estableciendo una conexión entre el final de S1 y el comienzo de S2, sin necesidad de un tercer elemento de unión.

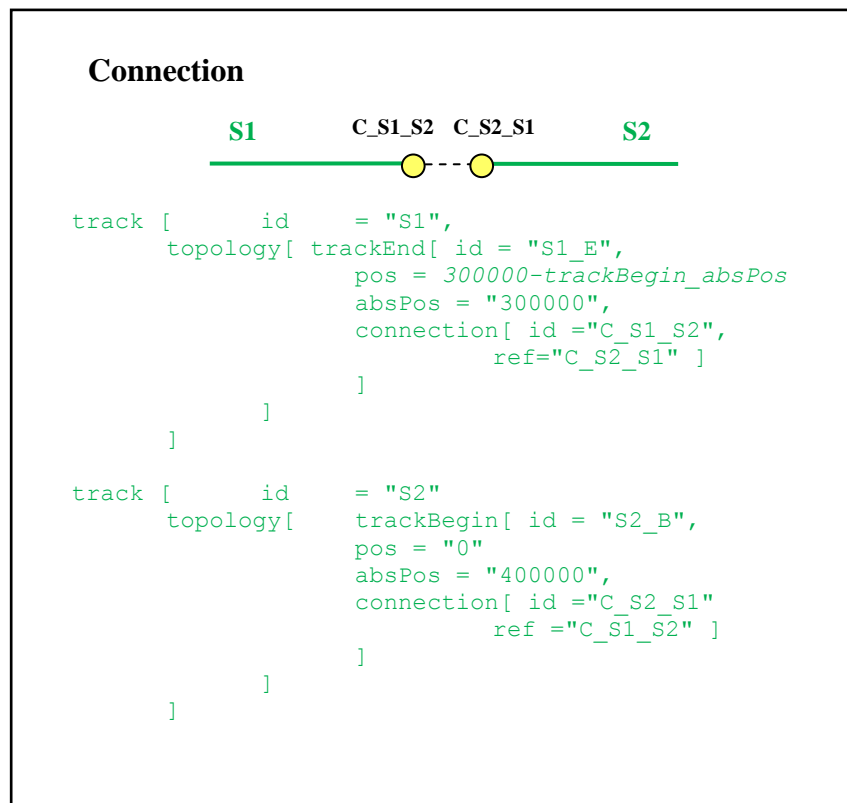


Figura 15: Ejemplo de conexión entre dos segmentos en RailML

Por una parte, dentro del elemento S1 del tipo `track`, el atributo identificador `id` adquiere el mismo valor que su homónimo dentro de los segmentos de UNISIG. Su comienzo estaría definido por un elemento `trackBegin` dentro de `topology`, con sus correspondientes atributos y conexiones con el resto de la línea. En cuanto a su extremo final, que es el que conocemos en este ejemplo, estaría identificado con el elemento `trackEnd`. Este incluye varios atributos. Por una parte, el identificador `id` que se ha establecido como el nombre del segmento seguido de "\_E" correspondiendo a "end". De este modo el `id` de S1 queda como S1\_E. Por otra parte, el atributo `absPos` que identifica el PK de este extremo, corresponde al atributo `kp1` de la conexión de este ejemplo en UNISIG, pero en este caso en metros, por lo que en vez de 300 adquiere el valor 300000. La posición relativa al comienzo del segmento, marcada por el atributo `pos`, correspondería a la resta del PK de este punto menos el PK del comienzo del segmento, ver (1).

$$pos(trackEnd) = absPos(trackEnd) - absPos(trackBegin) \quad (1)$$

Además, S1 tendría un elemento hijo del tipo `connection` con un `id` C\_S1\_S2 y una referencia a la conexión correspondiente en el comienzo del segmento S2, C\_S2\_S1.

El segmento S2 también contaría con su `id`, y en `topology`, su final estaría definido en `trackEnd` en función de cómo continúe la línea, mientras que `trackBegin` se encuentra condicionado por la conexión que aquí se define. De este modo, su `id` queda como "S2\_B", su posición relativa `pos` sería 0, ya que se trata del comienzo del segmento, y `absPos`, es decir, su PK corresponde a `kp2` de UNISIG en metros, es decir, 400000, y su conexión, llamada C\_S2\_S1 referencia al final de S1, C\_S1\_S2.

Se puede apreciar que al campo correspondiente a las coordenadas cartesianas en el Subset-112 no se le ha asignado ningún equivalente en el apartado actual, ya que este campo está destinado a las visualizaciones, de las que se ofrecen detalles más adelante.

### 7.3.3.2 Final de vía

Cuando nos encontramos en la topología con un final de vía, es decir con un tope que para el tren para evitar que se salga de la vía, se modela como indica la Figura 16.



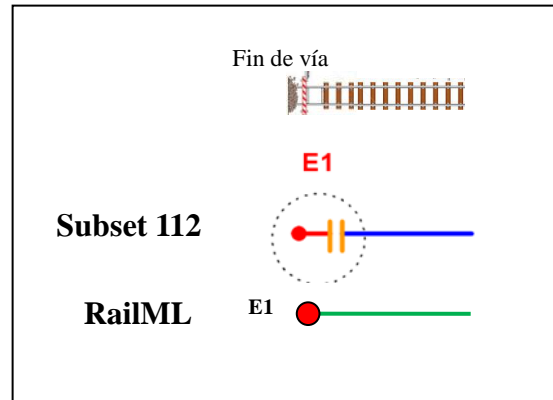


Figura 16: Fin de vía

En Subset-112 un final de vía con tope viene indicado con un elemento del tipo `end` conectado a un segmento mediante una conexión, mientras que en RailML se incluye en uno de los extremos de un segmento con el elemento `bufferStop`.

En el ejemplo detallado a continuación se modela un final de vía situado al comienzo del segmento S1, mostrando su estructura en el Subset-112 en la Figura 17, estando el elemento `end` E1 conectado al segmento de `id` S1 mediante a la conexión C\_E1\_S1 a la que referencia E1. Al ser el final un elemento adimensional, ambos PK de la conexión representada coinciden.

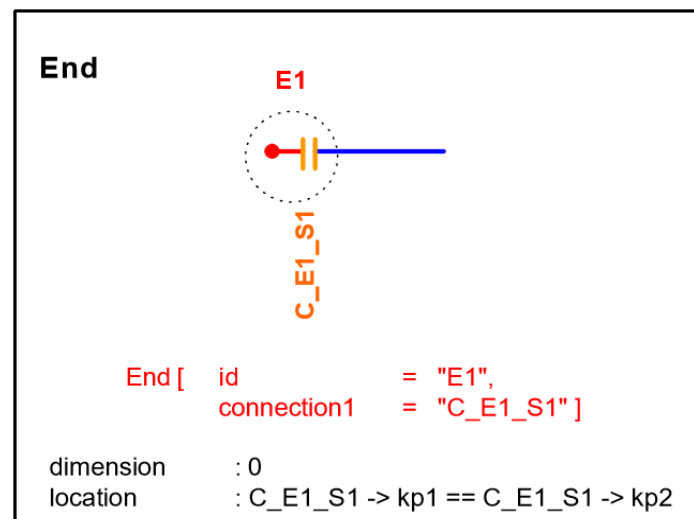


Figura 17: Ejemplo de final de vía en Subset-112

Para traducir el mismo ejemplo en RailML, como muestra la Figura 18, se establece en el elemento `track` de `id` S1, un comienzo de segmento `trackBegin` con `id` S1\_B (S1

begin) cuya posición relativa `pos` es nula al tratarse del comienzo del segmento y con posición en PK (`absPos`) igual al de la conexión `C_E1_S1` de la Figura 17. Al tratarse de un final de vía, en vez de establecerle a este extremo un elemento del tipo `connection`, se le establece un extremo del tipo `bufferStop`, con un `id` igual al del final en el Subset-112.

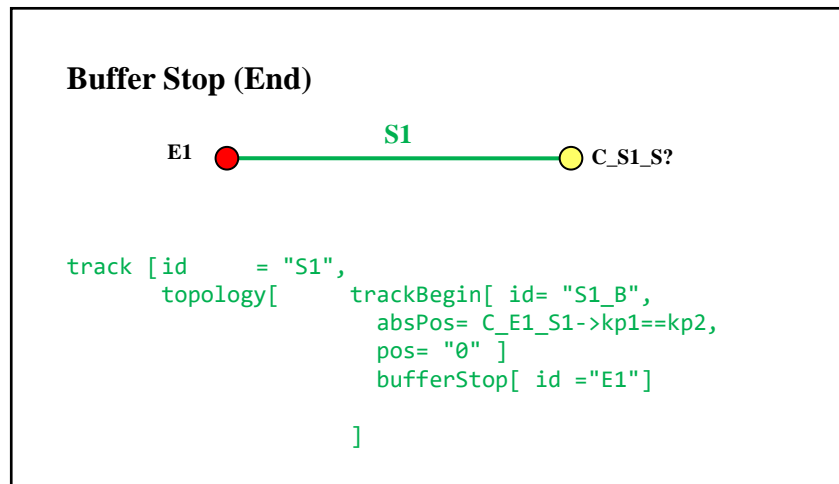


Figura 18: Ejemplo de final de vía en RailML

En este caso no se ha especificado los parámetros del extremo final de `S1` por falta de interés en la sección actual. Sin embargo, el elemento `trackEnd` es de obligada inclusión.

### 7.3.3.3 Conexión externa

Para la definición de una conexión con otra parte de la línea que el modelador no conoce, en Subset-112 se emplea el elemento `externalConnection` conectado a un segmento de la misma manera que se realiza en un final de vía. Mientras tanto, en RailML se modela como un final abierto o como una conexión normal en función de si se conoce o no los datos referentes a los elementos externos a los que se haya conectado.

En el ejemplo ofrecido a continuación se detalla la estructura de una conexión externa entre `TR1` y la vía externa `TR2`. En la Figura 19 se presenta esta conexión realizada con el formato de Subset-112, mediante el que se efectúa del mismo modo que un final de vía

(Ver Apartado 7.3.3.2), pero añadiendo dos atributos opcionales `extTrackside` y `extId` que referencian respectivamente a los `id` de la vía externa al documento actual TR2 y su correspondiente conexión externa E2.

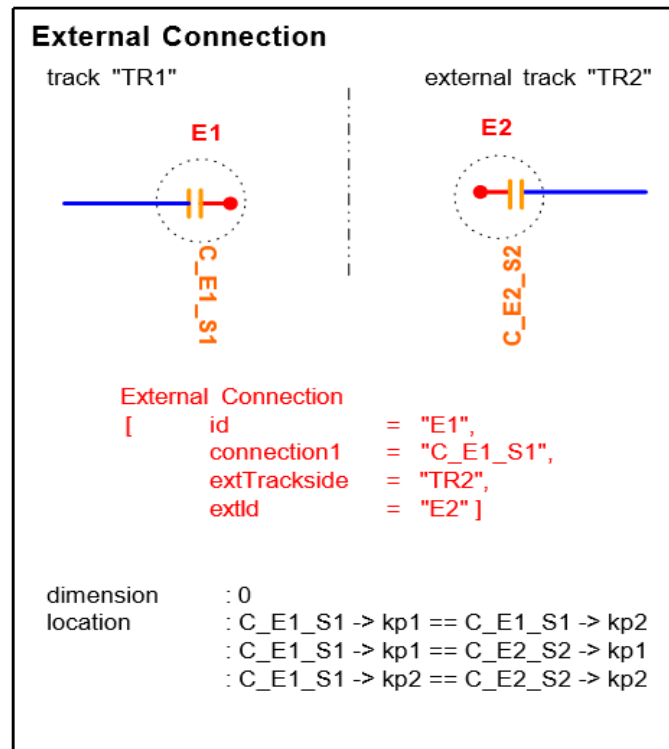


Figura 19: Conexión externa en Subset-112

Este ejemplo en RailML tiene dos posibles estructuras homólogas en función de si se conocen o no los datos correspondientes a TR2, es decir, a la vía externa.

En el caso de que estos datos no sean conocidos, esta conexión se realiza como se muestra en la Figura 20, igual que en el final de vía, pero en este caso se elige en el extremo del segmento, en este caso en el extremo final del tipo `trackEnd`, el elemento `openEnd` dentro de la estructura de elección, en vez de `bufferStop`. Este elemento representa un punto a partir del que no se incluye qué hay, bien por desconocimiento o por falta de relevancia de esa información. El elemento `openEnd` solo incluye el atributo identificador, que en este caso corresponde al mismo `id` "E1" de la Figura 19.

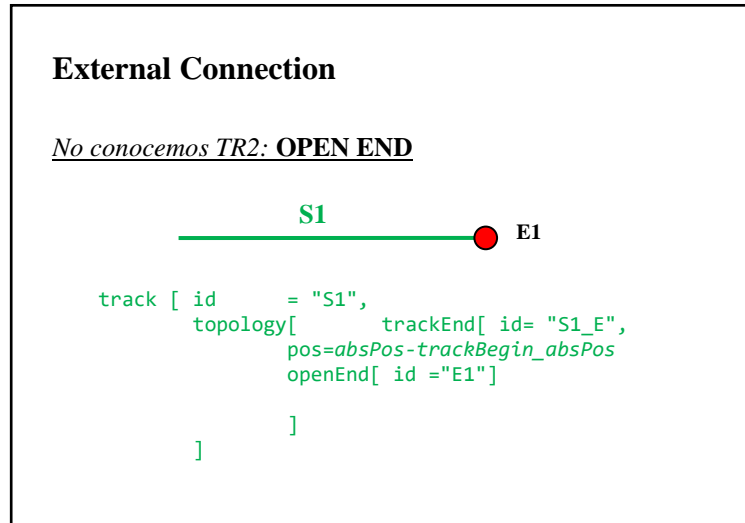


Figura 20: Conexión externa (open end) en Subset 112

En el segundo caso, es decir, si se conocen los datos de la vía externa TR2, se tratará esta estructura como una conexión normal de dos segmentos (Ver Apartado 7.3.3.1), incluyendo en las referencias de la conexión el id de la conexión de la vía externa, como indica la Figura 21.

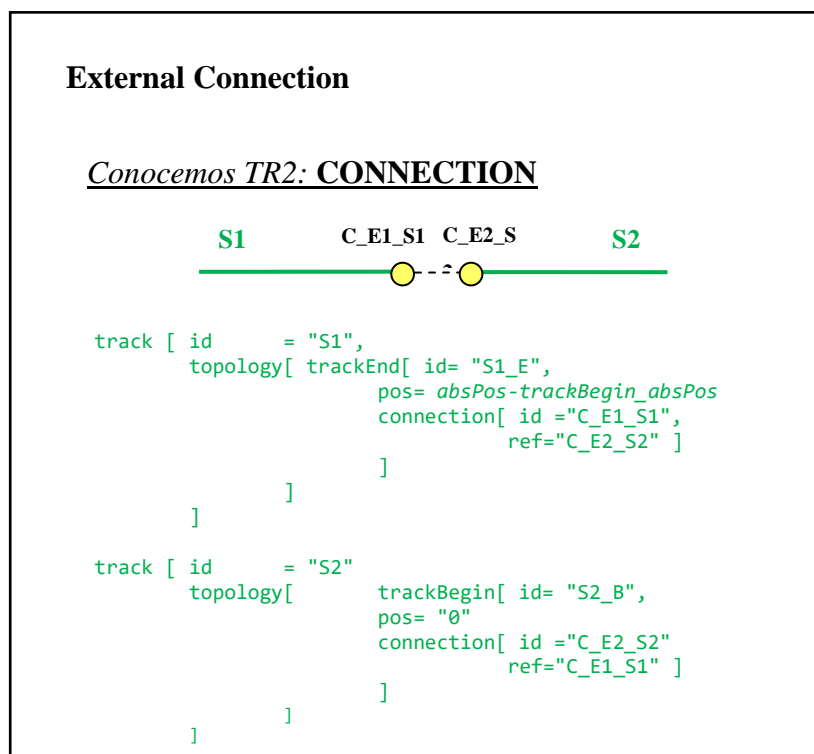


Figura 21: Conexión externa (connection) en Subset 112

### 7.3.3.4 Conexión entre tres segmentos (aguja)

La conexión entre tres segmentos se realiza mediante un desvío o aguja, que es un sistema mecánico que permite a un tramo de vía a continuar por dos tramos diferentes. cuya representación se muestra en la Figura 22.

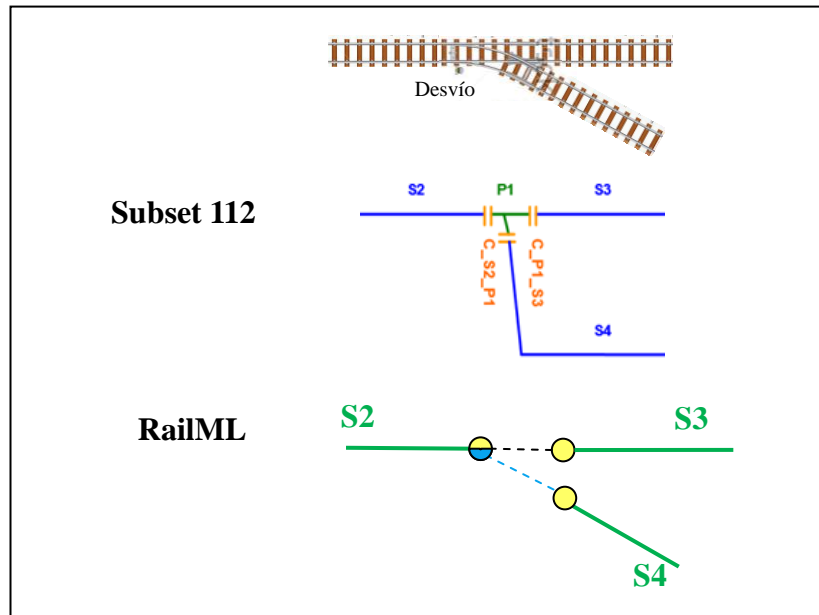


Figura 22: Aguja

Para la descripción de un desvío o aguja, se emplea en Subset-112 el elemento `point`, que supone un elemento adimensional que conecta el segmento que se deja atrás con otros dos, mediante tres conexiones que unen el punto en concreto con cada uno de los tres segmentos.

En RailML no existe un elemento que corresponda al punto que se acaba de mostrar, por lo que para modelar este caso se precisa unir el final de un segmento con el principio de otros dos. Sin embargo, los elementos de extremo de segmento `trackBegin` y `trackEnd` solo permiten seleccionar un único nodo, es decir, una única conexión con otro elemento. De esta manera, se enlazarían dos de los segmentos, el segmento originario, que es el que está en la `connection1` en el punto de UNISIG, con otro de los segmentos, por ejemplo con la siguiente conexión, la `connection2`, que representa al segmento que va a la izquierda. Quedaría por lo tanto, enlazar el segmento de origen con el segmento que va a la derecha, lo que se puede hacer mediante la inserción de un elemento `switch` como conexión en el segmento originario en su posición final.

En el ejemplo detallado a continuación extraído del Subset-112 de UNISIG describe la definición de una aguja que viene del segmento S2 y se divide en los segmentos S3 a la izquierda y S4 a la derecha, mediante el punto P1 que dispone de tres conexiones correspondientes a la unión de P1 con cada uno de los tres segmentos. Suponemos en este ejemplo que no se produce ningún cambio en la kilometraje, por lo que todos los puntos kilométricos de las tres conexiones son iguales entre ellos y de valor kp2, como queda indicado en la Figura 23.

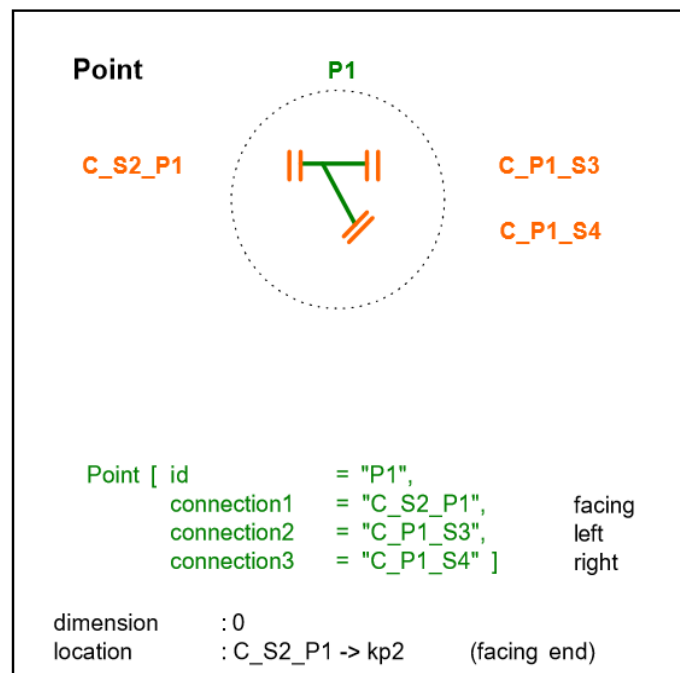


Figura 23: Aguja (punto) en Subset-112

El mismo ejemplo en RailML se modelaría como se muestra en la Figura 24, estableciendo una conexión normal entre el segmento S2 y S3 (Como se especifica en el apartado 7.3.3.1) junto con un switch en la posición final de S2, en este caso, el referente a kp2 especificado en la Figura 23. Este elemento tipo switch ha de referenciar al nodo de comienzo del segmento S4. Esta aguja tendrá un elemento de conexión que contará con orientación "both", ya que UNISIG no especifica si es o no una rama saliente o entrante, y con curso "right", ya que es la conexión que se encuentra a la derecha.

En el segmento S4 su extremo inicial se define igual que si se tratase de una conexión normal entre dos segmentos, pero referenciando a la conexión definida dentro de la aguja.

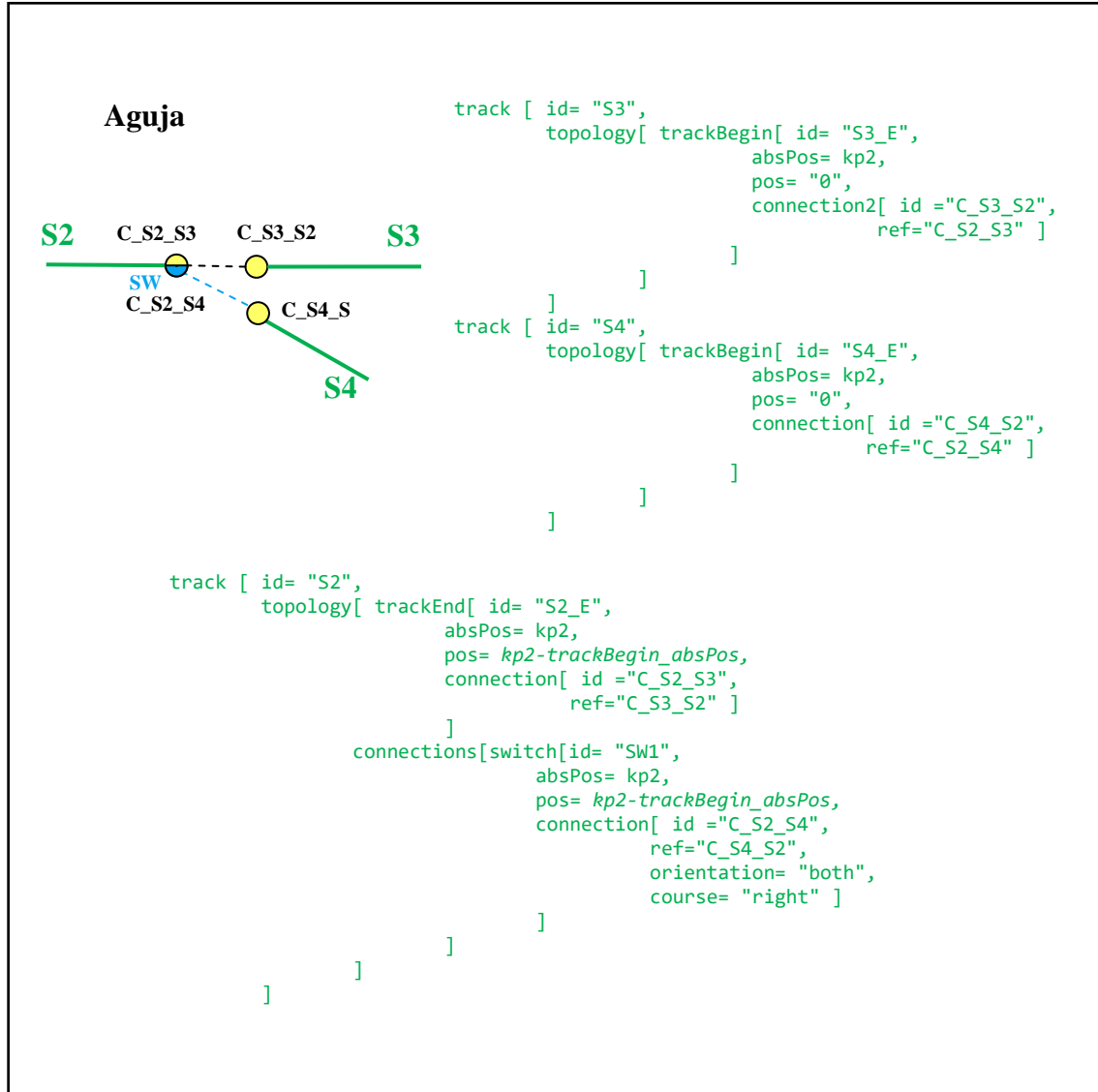
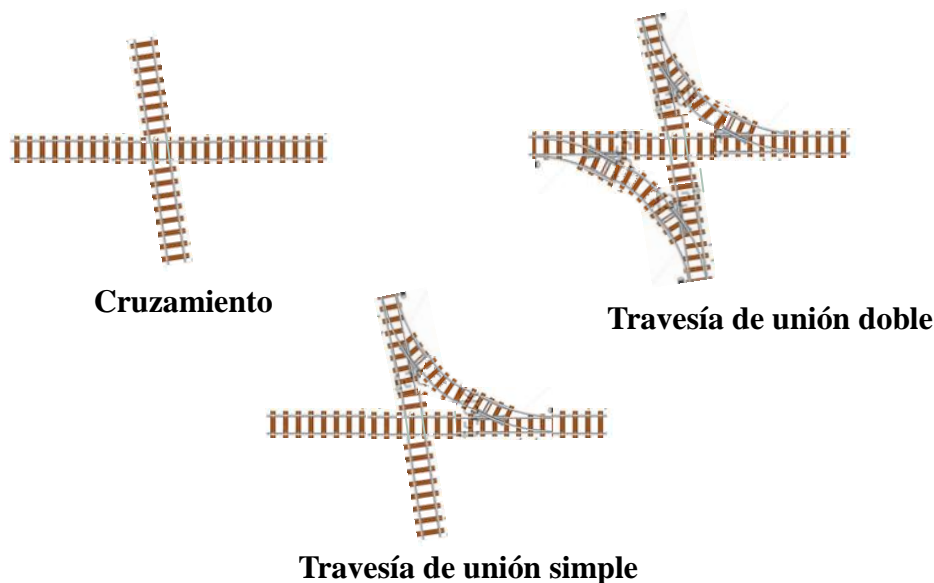


Figura 24:Aguja (punto) en RailML

### ***7.3.3.5 Cruzamientos, travesías de unión simple y travesías de unión doble***

Los cruzamientos, travesías de unión simple y de unión doble son cruces entre dos tramos de vía, que están diferenciados por los cambios que se pueden hacer entre ellos. En los cruzamientos, es un cruce meramente físico, ya que ningún cambio de vías está permitido. Por el contrario, en las travesías está permitido tanto el cruce como el cambio de vía. En las de unión simple solo se permite el cambio de vía desde una de las ramas entrantes al cruce, mientras que en el de unión doble se permite desde ambas ramas. Se representan estos tipos de cruces en la Figura 25.



*Figura 25 Cruzamiento, travesía de doble unión y de simple unión en Subset 112*

Según lo que establece el Subset-112, los tres casos anteriormente descritos se modelan del mismo modo, mediante la unión de dos puntos conectados entre sí mediante su primera conexión, es decir, en la dirección de cola, como muestra la Figura 26. Esto supone un problema, ya que los tres tipos de cruces quedan definidos del mismo modo, sin ninguna clase de diferenciación entre ellos, permitiendo los tres casos un comportamiento de travesía de doble unión.



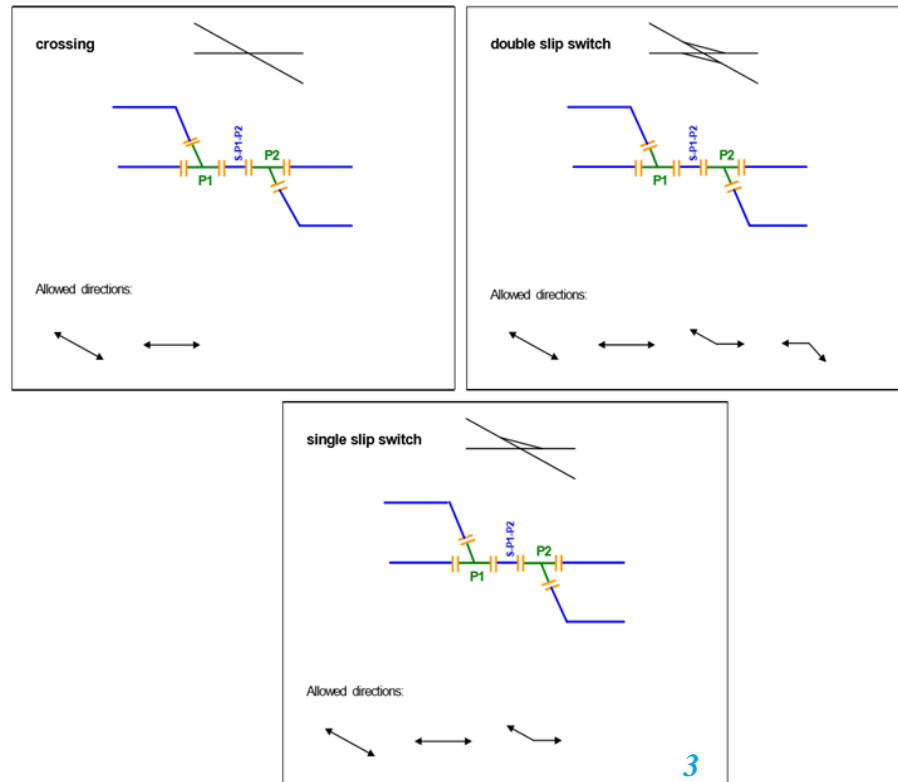


Figura 26: Cruzamiento (1), travesía de doble unión (2) y de simple unión (3) en Subset-112

En RailML, existe la posibilidad de modelar los tres casos como casos diferenciados. El cruzamiento se podría realizar como dos conexiones normales diferentes entre extremos de segmentos, dos a dos, o bien introduciendo una conexión de tipo `crossing`. El segundo y el tercero se pueden establecer con combinaciones de conexiones normales de los extremos y conexiones del tipo `switch`. Sin embargo, ante la falta de información sobre de qué tipo se trata por parte de UNISIG, y por lo tanto, ante la imposibilidad de diferenciar qué clase de cruce se ha de establecer, se modelarán los tres casos como si se tratase del caso más complejo y menos restrictivo, es decir, como una travesía de doble unión (Figura 26).

Por lo tanto, ya que en cada extremo de segmento sólo se permite la inclusión de un nodo y en este caso cada extremo precisa de dos conexiones, cada extremo incluido en esta travesía dispondrá además de una aguja. De este modo, como podemos ver en el ejemplo de la Figura 27 se modelarán como conexiones normales entre nodos aquellas que van "rectas", es decir, que no se producen mediante cambios de vía.

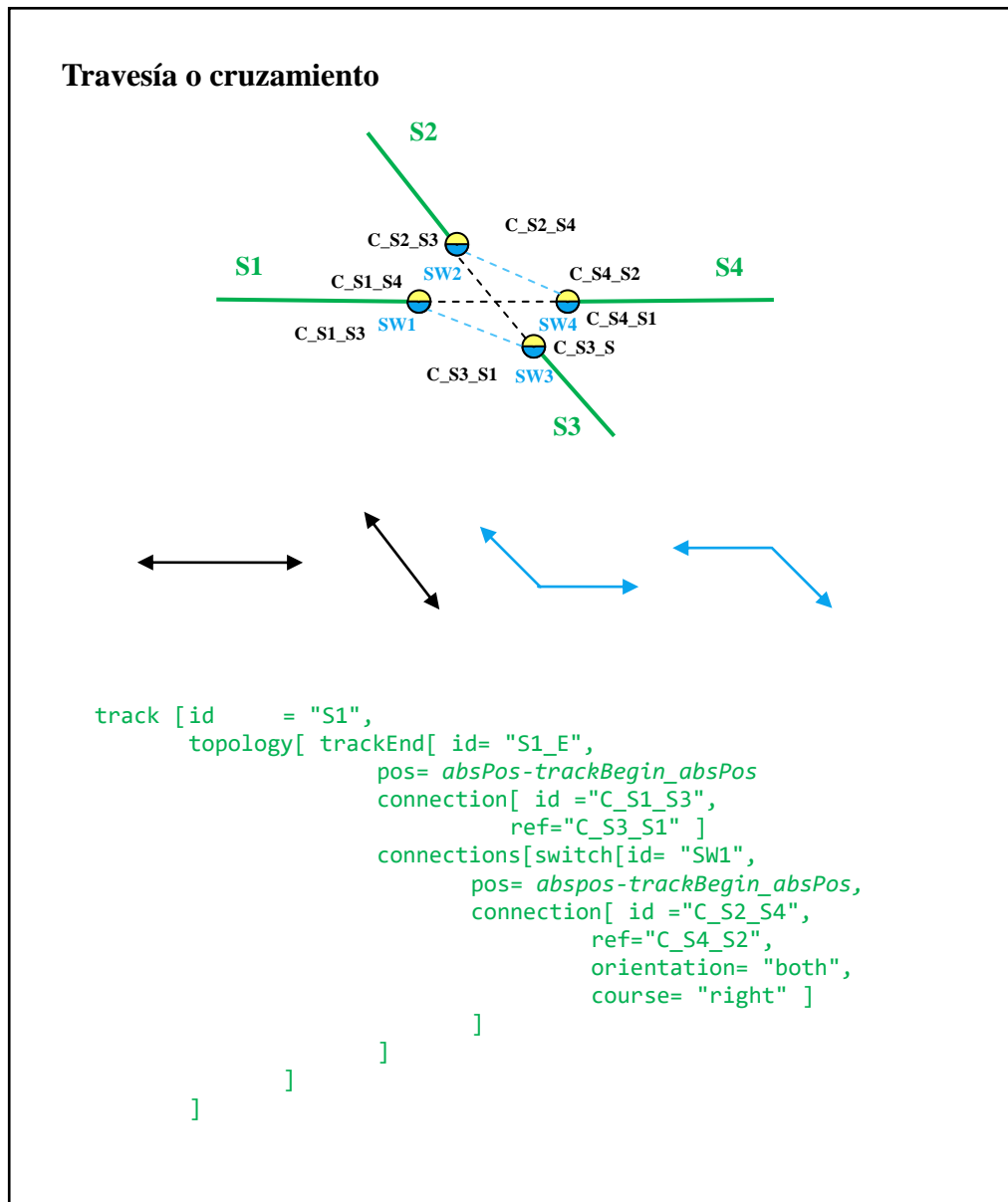


Figura 27: Travesía o cruzamiento en RailML

En el ejemplo de la Figura 27 estas conexiones corresponden a las marcadas en negro, que coinciden con aquellas entre los segmentos S1 y S4 y entre S2 y S3. En UNISIG estas conexiones corresponden a la unión entre ambas connection3, y entre ambas connection2, respectivamente. Se realiza esta asignación del mismo modo que en el apartado 7.3.3.1.

En cuanto al otro par de conexiones, aquellas que implican un cambio de vía, vienen marcadas en azul en el ejemplo y corresponden a las conexiones tipo `switch`, que se incluyen en los extremos de los cuatro segmentos. En este caso, se muestra únicamente la estructura escrita para el segmento S1, ya que los demás cuentan con la misma estructura.

En cada elemento `switch`, además del identificador y la referencia al elemento conectado, también se incluye la orientación, que en este caso será "both" en los cuatro segmentos, ya que en UNISIG ambas direcciones están permitidas. En cuanto al atributo `course`, que representa a qué lado de la vía principal se efectúa el desvío, en este caso es "right" para los segmentos en los que el desvío se produce a la derecha, en este caso para S1 y S4, y "left" para los restantes.

#### 7.3.3.6 Descarrilador

Un descarrilador (Figura 29) es un elemento que se coloca en la vía para hacer que el tren se salga de ésta, protegiendo así una zona en la que sería todavía más perjudicial que entrase el tren, por ejemplo una zona que se encuentre en obras, o donde pueda haber operarios trabajando en mantenimiento.



Figura 28: Descarrilador

Un descarrilador no es directamente un elemento topológico. Sin embargo, se incluye en este apartado debido a que en UNISIG se modela como una combinación de elementos topológicos. Como podemos ver en el ejemplo de la Figura 29, el Subset-112 lo define como un punto que conecta dos segmentos que son los que seguirían la línea, con un

segmento sin dimensión cuyo objetivo es conectar con un elemento de tipo fin de vía, que representa el descarrile.

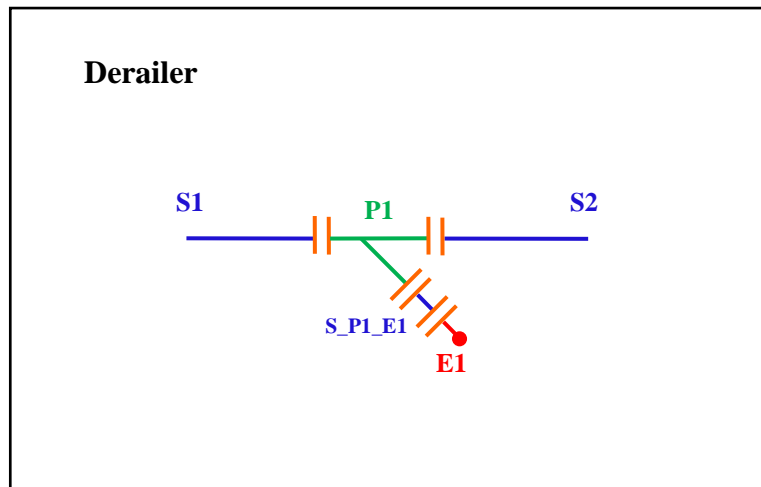


Figura 29: Descarrilador en Subset 112

Por su parte, RailML sí que dispone de un elemento que representa un descarrilador y que se puede incluir dentro de un segmento. Para este caso de trasponerlo desde el Subset-112 se colocará el descarrilador o bien en el final del segmento S1 o en el principio de S2, mediante la inclusión de un elemento del tipo `derailer` (descarrilador) que forma parte de los sistemas de operación y control (`ocsElements`), de los que se hablará más adelante. Se puede ver un ejemplo de descarrilador en RailML en la Figura 30.

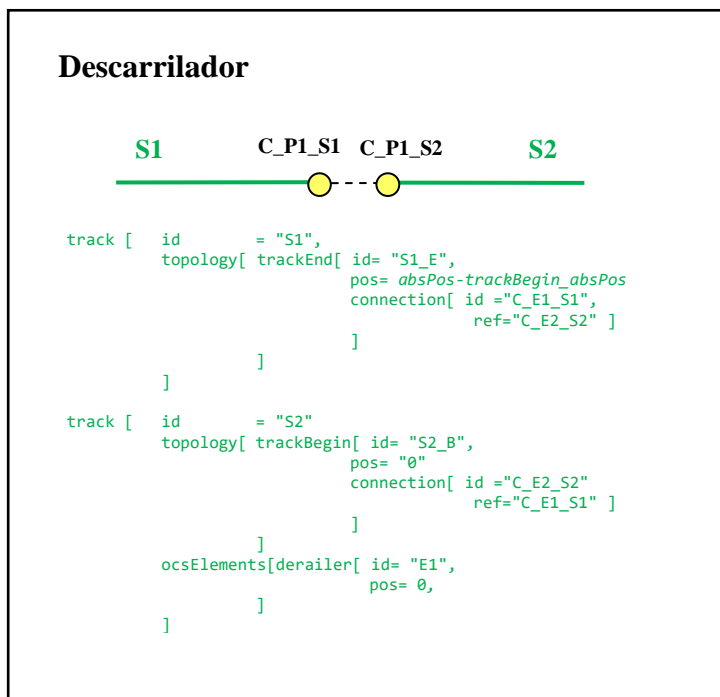


Figura 30:Descarrilador en RailML

De todos modos, aunque los descarriladores se encuentran definidos por UNISIG, en España están bastante en desuso, por lo que es una parte de la topología apenas utilizada.

## 7.4 POSICIONAMIENTO Y DIRECCIÓN

### 7.4.1 UNISIG (Subset-112)

Una vez la topología de la línea está completamente definida, ya se tiene una base sobre la cual comenzar a añadir elementos que se localizarán en diferentes segmentos. Estos elementos representarán diversos datos de interés que el tren se puede ir encontrando en su transcurso como puede ser diversas señales, cambios de electrificación, estaciones, pasos a nivel, etc.

Para marcar las ubicaciones de dichos elementos se utilizan **tres métodos de posicionamiento**, según la posición sea puntual, como puede ser por ejemplo en un cambio de velocidad, haga referencia a un trozo de línea como sucede en una estación, o se localice en todo un área formado por un conjunto de segmentos, como puede suceder para indicar las zonas ocupadas por otros trenes. Todos los datos relativos a posiciones en este formato se dan en kilómetros, ya que indican puntos kilométricos (PK), es decir la posición en relación a la kilometraje que se ha establecido en la vía.

- *Posicionamiento puntual*

Se emplea para la localización el tipo de datos `location`, que hace referencia a un segmento con su correspondiente `id` dentro del atributo `segmentid`, y le añade una distancia a mediante el atributo `offset` en kilómetros. De este modo, se define la posición de un punto situada a una distancia del comienzo del segmento marcada por el `offset`, como muestra la Figura 31. Es necesario incluir la referencia al segmento en el que se incluyen porque todos estos elementos se definen a parte de toda la topología.

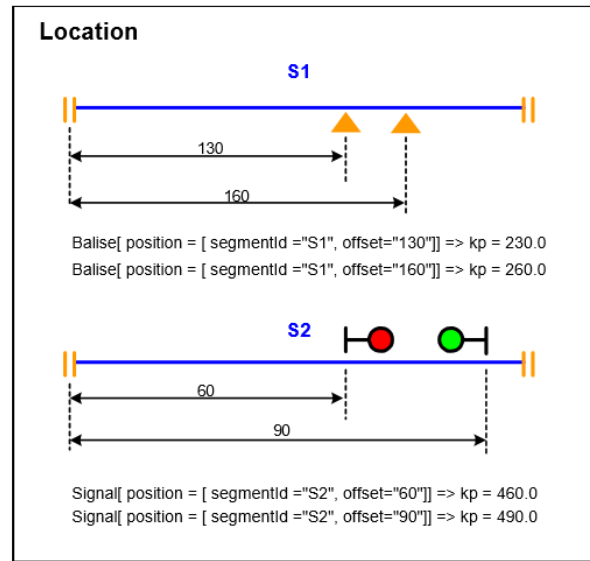


Figura 31: Posicionamiento puntual en Subset-112

- *Posicionamiento de línea*

Cuando la localización de un elemento no resulta suficiente con la definición de un punto, sino que es un elemento que ocupa un espacio de una determinada longitud de un segmento, se requiere el tipo de localización `lineLocation`, como puede suceder para definir una plataforma de una estación. En este caso se establece, además de la referencia al identificador del segmento en el que está localizado, dos elementos que representen una distancia al origen en kilómetros y que supongan el principio y final del tramo en el que este elemento se halla, como queda representado en la Figura 32. Estos dos atributos se denominan `startOffset` y `endOffset` y son de tipo `double`, es decir, valores numerales decimales.

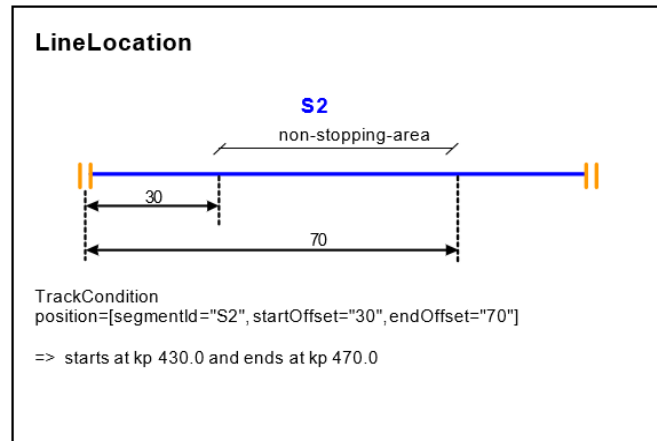


Figura 32: Posicionamiento de una línea en Subset-112

- *Posicionamiento de área*

Cuando un elemento no sólo ocupa una longitud en un segmento, sino que ocupa una zona en la que existen varios segmentos, como en una sección en la que no se permite parada, por ejemplo, en un túnel, su posicionamiento se define con la tipología *areaLocation*, que no es más que un conjunto de posiciones del tipo *lineLocation* comentado previamente, como se muestra en la Figura 33.

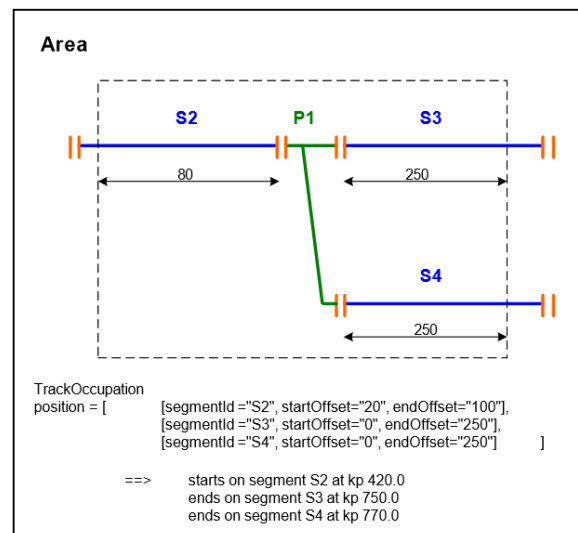


Figura 33: Posicionamiento de un área en Subset-112

Con estos tres últimos tipos se pueden definir todas las posiciones (en PK) de todos los elementos que se van a describir.



## 7.4.2 RailML

En el formato RailML, a la hora de posicionar elementos se utilizan diferentes atributos para su señalización. Todos aquellos elementos de la vía que precisen ser posicionados, cuentan con ciertos atributos que permiten posicionarlos.

No necesitan referenciar a ningún segmento, ya que estos elementos están incluidos dentro de la definición de cada segmento. Por otro lado, tienen todos ellos los siguientes atributos:

- **pos**: indica la posición que ocupa el elemento en relación con al comienzo del segmento al que pertenece. Este atributo es obligatorio y del tipo `tLengthM`, que define con un número decimal una distancia en metros.
- **absPos**: indica la posición que ocupa el elemento con su punto kilométrico. Este atributo es opcional y del tipo `tLengthM`, que define con un número decimal una distancia en metros.

A parte de estos atributos, aquellos elementos a los que para su posicionamiento no les basta con la determinación de un punto, sino que tienen cierta longitud, están dotados del atributo `length`, que define la distancia en la vía que el correspondiente elemento ocupa.

Además, en RailML todos los elementos tienen un elemento hijo denominado `geoCoord`, que guarda las coordenadas de un punto geográfico donde se localiza el elemento en cuestión. Esto podría ser de gran ayuda a la hora de posicionar elementos, pero el problema es que en España, las líneas ferroviarias no se encuentran geoposicionadas. El geoposicionamiento de las líneas españolas resultaría de gran utilidad para realizar un alto número de pruebas en diferentes campos pero los actuales proyectos propuestos para registrar estos datos pasan por un desembolso mínimo de 5 millones de euros.

### 7.4.3 Traducción del posicionamiento Subset-112 a RailML

Habiendo descrito los parámetros de posicionamiento de ambos formatos, podemos establecer una comparación entre ellos.

- *Posicionamiento puntual*

UNISIG utiliza el ya descrito tipo `location`, mientras que en RailML, basta con establecer la distancia al comienzo del segmento con el atributo `pos`, que equivaldría al `offset`, cambiando unidades de kilómetros a metros. Para establecer la `absPos`, es decir, el punto kilométrico en que se encuentra, se precisaría sumar el PK del comienzo del segmento más su distancia al comienzo (Figura 34).

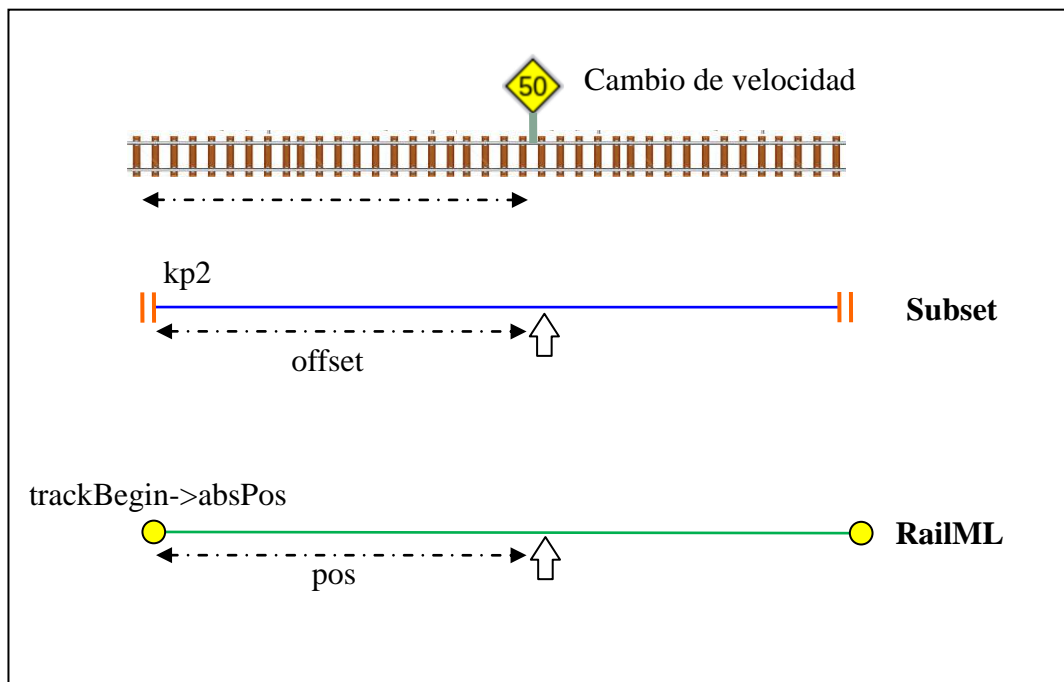


Figura 34: Posicionamiento puntual

- *Posicionamiento de línea*

Para un posicionamiento con longitud, por ejemplo, a la hora de posicionar un puente, en RailML se establecen dos distancias al comienzo del segmento que suponen el comienzo y final de la posición, mientras que en Subset-112, la diferencia de esas dos distancias, es decir, la longitud que ese elemento ocupa se establece con el atributo `length` (Figura 35).

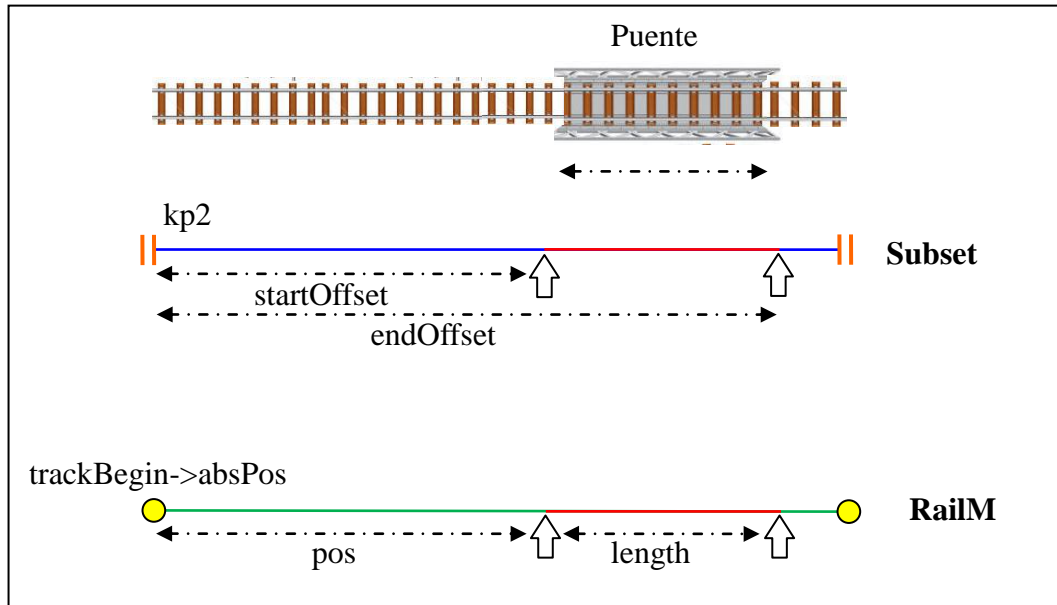


Figura 35: Posicionamiento de un elemento con longitud

- *Posicionamiento de área*

Subset-112 emplea el tipo `areaLocation`, mientras que según el formato RailML se debería incluir el mismo elemento en diferentes segmentos.

#### 7.4.4 Traducción de la dirección

Por otra parte, además de la posición, muchos elementos incluyen la dirección de la vía a la que dicha información se aplica, ya que a veces puede ocurrir que solo sea válido el cambio o el parámetro en uno de los sentidos de la vía, como puede ocurrir por ejemplo en un cambio de velocidad, ya que la velocidad permitida es función de la pendiente del trazado de vía, por lo que se aplicará una restricción de velocidad a cada sentido, ya que la pendiente es en un sentido positiva y en el contrario negativa.

Para definir el sentido de aplicación del cambio efectuado, UNISIG establece un tipo de datos denominado `directionEnum`, constituido por dos bits, y que cuenta con las correspondencias reflejadas en la Tabla 2.

Valor bits	Correspondencia
00	Dirección inversa
01	Dirección nominal
10	Ambas direcciones

*Tabla 2: Valores directionEnum*

El valor "Dirección nominal", representa la dirección con puntos kilométricos crecientes, es decir, desde el punto de inicio del segmento hasta su punto final. Por lo tanto, la "Dirección inversa" corresponde a la contraria.

Por su parte, en RailML también existe un atributo `dir` para especificar la dirección donde la mayoría de elementos son aplicables, pero en este caso es de tipo string y admite diferentes tipos de direcciones según qué tipo de dirección sea.

El tipo de dirección estándar de RailML (`tLaxDirection`) incluye cinco tipos de valores: "up", que correspondería con la llamada "Dirección nominal"; "down", correspondiente a la "Dirección inversa", "both" para ambas direcciones simultáneamente; y "none" y "unknown", cuando la restricción no existe o no se conoce respectivamente. Estas dos últimas no se emplean por UNISIG.

Existe un segundo tipo en RailML, el de dirección estricta, para cuando la dirección es más restrictiva, es decir, cuando solamente se puede incluir uno de los dos sentidos: "up" o "down". Para este caso, si en un documento con el formato de UNISIG se establece un elemento con ambas direcciones activadas, se debería crear en RailML el elemento por duplicado, y aplicarle a cada uno de ellos uno de los sentidos.

En el caso que sea válido en ambas direcciones habría que crear dos elementos de cambio de galíbo en RailML, cada uno para definir el cambio en una dirección.

## 7.5 ELEMENTOS EN LA VÍA

Una vez está definida la topología de la infraestructura bajo estudio, y se han definido modos de posicionar elementos sobre ella, se puede pasar a la definición de los mismos.

Con elementos de la vía están referidos todos aquellos elementos físicos que se le pueden añadir a la vía sobre su topología, y que aportan datos que pueden resultar necesarios ante la realización de pruebas de señalización ERTMS. Se engloban en este apartado por lo tanto todos los datos acerca de los diferentes sistemas de señalización, partes especiales de la vía, como zonas de retroceso, túneles, zonas con restricciones de velocidad, etc.

Esta información permite simular todas aquellas condiciones a las que estaría sometido el tren en una infraestructura real, simulando todos los elementos que afectarían a su marcha y al control del tráfico. Podemos ver algunos ejemplos de estos elementos en la Figura 36.

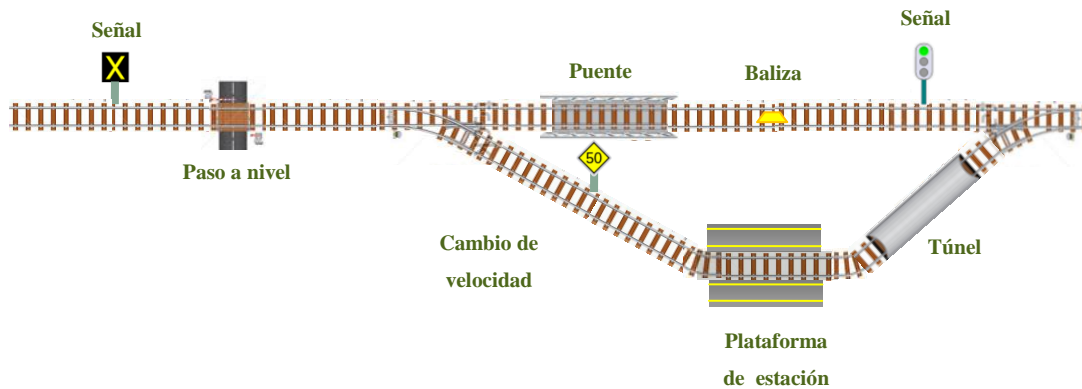


Figura 36: Elementos de vía

En el protocolo establecido por UNISIG, todos los elementos de vía se encuentran recogidos bajo la lista `trackElements`, que se encuentra al mismo nivel en el que se encontraba la topología.

Por su parte, en RailML estos elementos se encuentran encerrados en diferentes elementos, estando la mayoría de ellos definidos dentro de cada segmento. Los elementos definidos dentro del segmento se diferencian en dos grandes grupos: aquellos

considerados "elementos físicos" (`trackElements`), que son elementos "tangibles" que no cambian en la vía, como túneles o restricciones de velocidad y los "sistemas de operación y control" (`ocsElements`) que se refieren a aquellos encargados de la señalización y el control de tráfico, como balizas o detección de trenes.

### 7.5.1 Elementos físicos

Se comenzará haciendo un encuadre de formato para los elementos de vía considerados como "físicos", es decir aquellos tangibles, cuyo mensaje siempre es el mismo y no se altera según las condiciones del tráfico o del material rodante. Todos estos elementos se encuentran en ambos formatos bajo el elemento correspondiente a los elementos de vía (`trackElements`)

#### 7.5.1.1 Cambio de gálibo

El gálibo se refiere a las medidas máximas que pueden tener los vagones de tren, para pasar con distancias suficientes tanto verticales como laterales por túneles, puentes y demás tramos con limitación de espacio. Cuando se produce un cambio en estas características, es decir, cuando se cruza un tramo de vía con condiciones diferentes que permitan distancias y medidas diferentes, es necesario registrarlo e indicar cuál es el nuevo gálibo permitido.

Para definir la **traducción**, en UNISIG este cambio de gálibo se emplea el elemento `lineGaugeChange`, mientras que en RailML el correspondiente elemento es `clearancGaugeChange`.

El cambio de gálibo en Subset-112 cuenta con cuatro atributos de los que a continuación se establecen sus traducciones en RailML.

- Un elemento identificador con el que cuentan ambos formatos denominado `id`, que en ambos formatos es de tipo string por lo que la correspondencia resulta unívoca.
- Posicionamiento puntual del tipo `location` (apartado 7.4.3).
- Dirección de sentido estricto (apartado 7.4.3).
- Tipo de gálibo que se adopta a partir de este punto.

Existen diversos gálíbos estandarizados para diferentes anchos de vía, cada uno de los cuales permite diferentes cargas por eje y velocidades, recogidos en las Especificaciones Técnicas de Interoperabilidad relacionadas con la infraestructura, publicadas por la Agencia Ferroviaria Europea[18].

Según el tipo de ancho, están definidos unos gálíbos estándar u otros. En el ancho estándar europeo, el denominado UIC, de 1435 mm de ancho, se tienen definidos los gálíbos llamados GA, GB, GC como aquellos más comunes, y G1 y G2, como gálíbos internacionales para especificaciones no determinadas (Figura 37). En España se tienen estos gálíbos para las líneas de AVE, y otros estándares nacionales para el resto de líneas, ya que tienen un ancho de vía diferente [19][18].

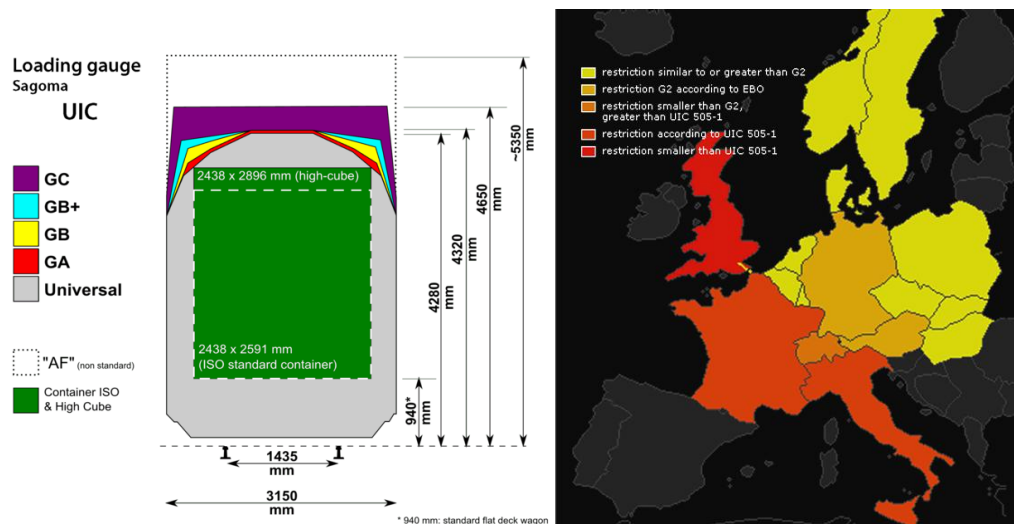


Figura 37: Gálíbos internacionales.[19]

En el Subset-112 se definen estos gálíbos mediante 8 bits, cada uno correspondiente a un gálíbo diferente, pudiendo así tener varios activados al tiempo[20].

Bit activo	Código
xxxxxxx1	G1
xxxxxx1x	GA
xxxxx1xx	GB
xxxx1xxx	GC
1111xxxx	Bits libres para la inclusión de nuevos gálibos

Tabla 3: Gálibos Subset-112

Los datos ofrecidos en la Tabla 3 están en orden de restricción decreciente, es decir, cuanto más a la derecha esté el bit activo es más restrictivo el gálibo.

Por su parte, en RailML, solo se incluye uno de los gálibos mediante su correspondiente código alfanumérico descrito anteriormente (Por ejemplo, GA), que correspondería con el menos restrictivo. Todos los posibles códigos que recoge RailML están recogidos en su documentación en la lista de código TrainClearanceGauge.xml[13].

En el caso de que en el Subset-112 se hayan validado varios gálibos diferentes, se incluirá en RailML únicamente el menos restrictivo, ya que el que sea menos restrictivo incluye a todos los demás que estén incluidos.

### 7.5.1.2 Cambios en el voltaje

Actualmente, los trenes son traccionados gracias a la acción de motores eléctricos. Para alimentar estos motores se hace uso de las catenarias, que son líneas de cable instalados por encima de la vía para proveer de corriente al tren.

Los voltajes empleados en la alimentación de los sistemas de tracción en la infraestructura ferroviaria española son 25 kV en corriente alterna o 3 kV en continua, con excepciones de líneas que operan a 1,5 kV[8].

Por lo general en España, por lo tanto, no se suelen producir cambios en el voltaje, únicamente se puede producir en corriente continua. En cambio, esta situación sí se puede dar en otros países o al pasar de un país a otro y cambiar de voltaje e incluso de frecuencia de alimentación, como se puede ver en la Figura 38. Es por esto que se precisa indicar dónde y cómo ocurren estos cambios de voltaje.



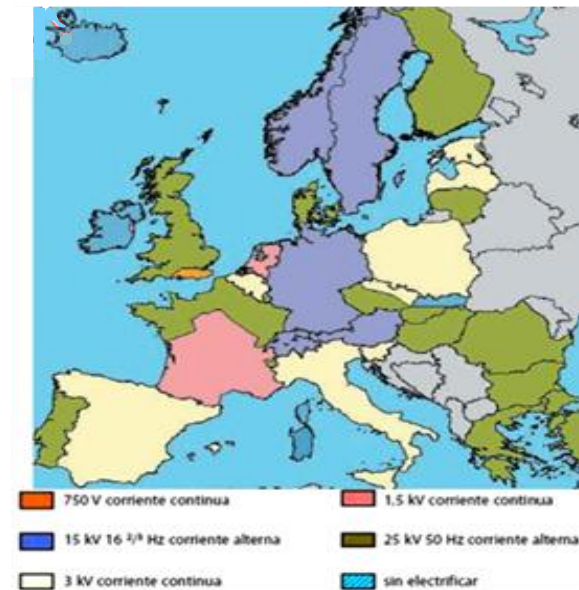


Figura 38: Voltajes usados en Europa (simplificado)[2]

Para este fin, el Subset-112 emplea el elemento `voltageChange` que define un cambio de esta clase y cuyos atributos tienen sus homólogos en el elemento `electrificationChange` en RailML.

Para la **traducción**, se tienen en cuenta los siguientes atributos:

- Los elementos de ambos formatos comparten los elementos **identificadores**, por lo tanto su relación es directa.
- En cuanto a la **localización** y el **dirección** de este elemento, están definidos respectivamente por un posicionamiento puntual y una dirección estricta, definidos en el apartado 7.4.3.
- Por otra parte, este cambio incluye la **información acerca del voltaje** al que se va a cambiar. UNISIG solo da pie a ciertos valores que se identifican como estándares, que corresponden con todos aquellos mostrados en el mapa de la Figura 38, más líneas con voltajes de 600 y 750 V DC, usados fundamentalmente en líneas de tranvía y metro. El valor de este voltaje está recogido por UNISIG en un atributo definido por 4 bits como se muestra en la Tabla 4.

Número	Bit activo	Voltaje
0	0000	No se incluye en ningún valor estándar
1	0001	AC 25 kV 50 Hz
2	0010	AC 15 kV 16.7 Hz
3	0011	DC 3 kV
4	0100	DC 1.5 kV
5	0101	DC 600/750 V
6-15	0110-1111	Valores de reserva

Tabla 4: Valores de voltajes UNISIG[20]

Para encajar estos valores en RailML se emplean dos de los atributos de `electrificatioChange`, que son `voltage`, que encuadra el voltaje en voltios como valor numérico decimal, y `frequency` que representa la frecuencia en Hz para la corriente alterna como número decimal también.

- Por último, se establece en UNISIG que si el voltaje corresponde a un valor estándar se especifique el **identificador del país** al que pertenece, pero más tarde UNISIG consideró la *falta de relevancia de dicha información*.

### 7.5.1.3 Cambio de categoría de carga por eje

Las vías de tren, según características de su infraestructura, como peraltes, radios de curvatura o pendiente, permiten unas determinadas cargas máximas por eje del tren, ya que superadas estas cargas, la infraestructura podría deteriorarse o incluso llegar a fracturarse.

Sin embargo, el único factor que puede deteriorar la infraestructura o que puede crear una situación de peligro, no es la carga por eje, también influye a qué velocidad se transporta dicha carga. De este modo, hay veces que el límite de velocidad varía según la carga aplicada.

Para esta limitación, UNISIG emplea en el Subset-112 el elemento `axleLoadCatChange`, que identifica un cambio en estas restricciones de velocidad asociadas a las cargas. En RailML, en cambio, *no existe un elemento de cambio de velocidad debido a la carga*, así que quedaría registrado como un cambio de velocidad

normal en el elemento de cada tramo `speedChanges`, contando con los siguientes atributos de UNISIG, y haciendo así efectiva la traducción.

- Ambos elementos de ambos formatos comparten el atributo **identificador**.
- **Dirección estricta.**
- **Posicionamiento puntual.**
- Además, `axleLoadCatChange` recoge un atributo denominado `trainDelay`, de tipo booleano, es decir, de solo un bit. Este atributo describe **a partir de qué punto del tren la restricción de velocidad es efectiva**. Si tiene valor 0, se aplica la restricción cuando el final del tren ha pasado el punto de cambio de velocidad, y si tiene valor 1, cuando el comienzo del tren pasa este punto. Este parámetro se incluye en RailML en el atributo `trainRelation`, que adquiere los valores 'headOfTrain' o 'endOfTrain' para el principio y el final del tren respectivamente. Por defecto, este atributo aplica el cambio al comienzo del tren cuando la velocidad del tren disminuye con el cambio, y al final del tren cuando supone un aumento de velocidad.
- Lista de datos que recoge UNISIG referentes a distintas **categorías de carga por eje**. En cada elemento de la lista de datos, del tipo `axleLoadSpeed`, se contienen dos elementos. Uno de ellos es un entero que hace referencia a las categorías de carga por eje, que UNISIG clasifica en 7 bits (128 valores), según la numeración mostrada en la Tabla 5.

Nº	Categoría
0	A
1	HS17
2	B1
3	B2
4	C2
5	C3
6	C4
7	D2
8	D3
9	D4
10	D4XL
11	E4
12	E5
13-127	Valores extra

Tabla 5: Clasificación de categorías de carga por eje por UNISIG[20]

Cada una de las categorías expuestas corresponde a una carga diferente recogidas en las Especificaciones Técnicas de Interoperabilidad para la Infraestructura[21].

- Una **restricción de velocidad** para cada una de las categorías recogidas, como ocurren en `axleLoadCatChange`. Esta velocidad se recoge en 7 bits, es decir en 128 valores, de los cuales se utilizan los 121 primeros (0-120), para registrar velocidades entre 0 y 600 km/h en intervalos de 5km/h. A mayor nº identificador de las clases indicadas en la Tabla 5, mayor carga lleva el tren y menor velocidad puede adquirir

Esta información es leída por el tren, quien compara su propia categoría con las que ofrece la vía y adquiere la restricción de velocidad correspondiente a partir de ese punto.

En el caso de RailML, se añadiría un cambio de velocidad `speedChange` para cada tipo de tren, añadiendo la categoría correspondiente como un atributo extra, y la velocidad (multiplicada por 5 para su conversión a velocidad real) en el atributo `vMax`, que registra la velocidad máxima en km/h.

#### 7.5.1.4 Cambios estáticos de velocidad

Cuando se producen otras restricciones de velocidad que no están sujetas a la carga del tren, se aplica el cambio estático de velocidad. En UNISIG se recoge bajo el elemento `staticSpeedChange`, y se corresponde de nuevo con el elemento `speedChange` de RailML. De este modo, los atributos correspondientes al identificador, posicionamiento puntual, dirección de aplicación estándar y el punto del tren desde donde se aplica, se traducen del mismo modo que se hace con los cambios de velocidad por carga por eje, detallado en el apartado 7.5.1.3.

Este elemento de UNISIG recoge múltiples restricciones de velocidad según diferentes factores. Para cada cambio de velocidad habrá que definir un elemento `speedChange` diferente, aunque todos con la misma posición, dirección y punto del tren donde se hace efectivo. En cambio, habrá que efectuar algún cambio para cada id, ya que deben ser únicos.

El primer cambio de velocidad que se incluirá es la **restricción por defecto**, sin tener en cuenta ninguna clasificación del tren. Esta se aplicará cuando el tren no pertenezca a ninguna categoría en especial. El único atributo que se le añade aparte de los mencionados anteriormente es aquel que define la velocidad. La velocidad por defecto está en el atributo `speed`. Esta velocidad se recoge en 7 bits, es decir en 128 valores, de los cuales se utilizan los 121 primeros (0-120), para registrar velocidades entre 0 y 600 km/h en intervalos de 5km/h. Este valor en RailML se multiplicaría por 5 para su conversión a velocidad en km/h y se incluiría en el atributo `vMax` de `speedChange`.

En el caso de que se especifique la **categoría del tren** se le podría permitir adquirir otra velocidad diferente, que puede ser mayor de la establecida por defecto. Estas velocidades se incluyen en un vector de elementos `trainCategorySpeed`, cada uno de los cuales recoge una categoría de tren y su correspondiente restricción de velocidad.

En este elemento existen dos tipos de clasificaciones en UNISIG. Una hace referencia al tipo de carga que se lleve, es decir si el tren lleva pasajeros o mercancías. La otra en cambio, clasifica los trenes según la máxima insuficiencia de peralte que permitan. El peralte es la pendiente transversal de la vía que se habilita en las curvas para contrarrestar la fuerza centrífuga que sufren los trenes en ellas. Por lo tanto, la insuficiencia de peralte

es la diferencia entre el peralte que tiene realmente la vía en una determinada curva y el peralte que requiere el tren en esa curva a una determinada velocidad.

Dentro de cada categoría UNISIG establece varios atributos:

- **qualifier** de valor entero, que determina dentro de qué tipo de clasificación entra esta categoría. En el caso de que tome valor 0, se aplica una clasificación según la insuficiencia de peralte, mientras que si toma valores 1 o 2 se usa la clasificación según el tipo de carga[20].
- **speedDiff**, que es el valor de la restricción de velocidad que adquiere el tren si pertenece a dicha clase. Al igual que **speed**, los valores toman velocidades entre 0 y 600 km/h a intervalos de 5km/h y se incluyen por lo tanto multiplicadas por 5 en el atributo **vMax** de **speedChange** en RailML.

En cuanto a la clase a la que corresponde cada categoría de la lista, hay dos atributos diferentes en Subset-112 para cada tipo de clasificación.

- Si el calificador **qualifier** indica que se trata de una categoría relacionada con la *insuficiencia de peralte* (valor 0), el indicador de categoría en UNISIG, que corresponde a un entero de 4 bits se incluye en el atributo **cantDeficiencySSPCategory**, correspondiendo cada valor con una insuficiencia de peralte diferente [13].
- Si por el contrario, se indica que la clasificación se da por el *tipo de carga*, el indicador de categoría en UNISIG, que corresponde a un entero de 4 bits se incluye en el atributo **trainCategory**. Según el valor que tome este atributo se puede seleccionar entre trenes de pasajeros y trenes de mercancías con frenos en posición "P" y "G" [13]. Estas dos posiciones indican dos tipos de frenado con diferente fuerza, siendo la posición "G" la que consigue frenar una masa mayor en menos tiempo.

Todas estas categorías vienen recogidas en RailML en el atributo **etcsTrainCategory** de **speedChange**, que incluye 15 bits, cada uno correspondiente a una categoría diferente y activo en caso de corresponderse con esa categoría[22]. La correspondencia de las categorías de ambos formatos se muestra en la Tabla 6.

Subset-112	RailML	
cantDeficiency	etcsTrainCategory	Insuf. de
SSPCategory	(valores de los bits)	peralte (mm)
0	xxx xxxx xxxx xx1x	80
1	xxx xxxx xxxx x1xx	100
2	xxx xxxx xxxx 1xxx	130
3	xxx xxxx xxx1 xxxx	150
4	xxx xxxx xx1x xxxx	165
5	xxx xxxx x1xx xxxx	180
6	x1x xxxx xxxx xxxx	210
7	xxx xxxx 1xxx xxxx	225
8	xx1 xxxx xxxx xxxx	245
9	xxx xxxx xxxx xxx1	275
10	xxx xxx1 xxxx xxxx	300
trainCategory	etcsTrainCategory	Tipo de carga
0	xxx xx1x xxxx xxxx	Mercancías P
1	xxx x1xx xxxx xxxx	Mercancías G
2	xxx 1xxx xxxx xxxx	Pasajeros

Tabla 6: Categorías para velocidades en RailML y UNISIG

### 7.5.1.5 Cambios de pendiente

Otro punto importante a la hora de describir una infraestructura son las pendientes que afectan a esta, ya que de las pendientes de una vía dependen muchos otros parámetros como velocidades máximas, cargas máximas, frenos, etc.

El elemento de nombre `gradientChange`, representa un cambio de pendiente tanto en Subset-112 como en RailML. Se representa como un punto a partir del cual la pendiente de la vía cambia de valor.

Esta **traducción** es bastante directa, contando con las siguientes correspondencias de atributos.

- Elemento **identificador** en ambos formatos.
- El **posicionamiento** es puntual.

- **Dirección** estricta.
- Atributo con el que definen la nueva **pendiente** en valores de tanto por mil (‰), que es la unidad en la que comúnmente se representan las pendientes en la industria ferroviaria. Esta unidad representa el número de unidades de longitud que asciende (o desciende en caso de una pendiente negativa) el terreno por cada mil unidades en la horizontal, como podemos ver en el ejemplo de la Figura 39.

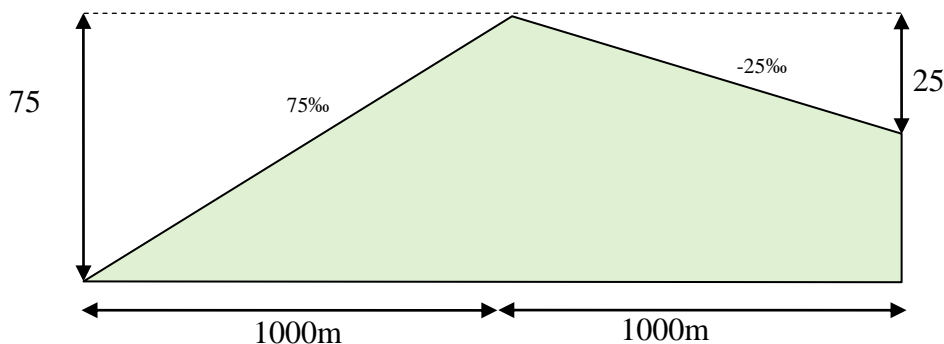


Figura 39: Ejemplo de Pendientes

Esta nueva pendiente viene recogida en Subset-112 en el atributo `gradient`, cubriendo las pendientes entre +254‰ y -254‰ con números enteros y siendo los valores positivos correspondientes a subidas y los negativos a bajadas.

El mismo atributo está incluido en RailML bajo el nombre `slope`, que también supone un valor numérico con signo representando la pendiente en tanto por mil, aunque esta vez en valores decimales.

#### 7.5.1.6 Condiciones de la vía

Las condiciones de la vía corresponden con un elemento que cubre circunstancias muy diferentes a las que puede verse sometida un tramo.

En este caso, las condiciones a las que vemos sometida la vía son las siguientes:

- *Área sin paradas* : área donde está prohibido parar, como puede ser un túnel o un puente
- *Activación de la bocina*: áreas en las que la activación de la bocina como señal sonora es obligatorio.



- *Zona sin electrificación*: zona en la que no hay electrificación, bien por encontrarse desactivada o por no existir catenaria en esa área. En estas zonas se deben bajar los pantógrafos del tren, es decir, los recolectores de energía eléctrica que entran en contacto con la catenaria.
- *Zona sin recepción de radio*: zona donde los receptores de radio no reciben ninguna señal. Se debe dejar de comprobar en estas áreas la señal que se emite para hacer saber que todo está correcto, ya que se está incomunicado y puede llevar a problemas al hacer creer que ha habido algún incidente.
- *Apagado de la principal fuente de energía*: zona que requiere del apagado de la misma.
- *Desactivación de diferentes frenos*:
  - **Frenos regenerativos**. Son aquellos que devuelven la energía de frenado a la catenaria en forma de energía eléctrica.
  - **Frenos de corriente de Foucault** (Eddy current brake). Usan este tipo de corrientes para el frenado.
  - **Zapatas magnéticas**. Emplean electromagnetismo en la frenada.

En ambos formatos este elemento está recogido bajo el nombre `trackCondition`. Sea de la condición que sea, este elemento contiene en ambos formatos:

- Un **identificador**.
- **Posicionamiento** de tipo área.
- El **tipo** de condición de vía dado por el atributo `type`. La diferencia es que según UNISIG este atributo viene dado por un número entero de 4 bits, mientras que en RailML viene dado por un string, entre los que hay unos valores preestablecidos, aunque se pueden añadir nuevos. La equivalencia entre ambos formatos se muestra en la Tabla 7.

Descripción	Valor type UNISIG	Valor type en RailML
Área sin paradas	0000	nonStoppingArea
Área sin paradas por un túnel	0001	nonStoppingArea
Activación de la bocina	0010	Other
Zona sin electrificación. Se baja el pantógrafo	0011	lowerPantograph
Zona sin recepción de radio	0100	radioHole
Zona de estanqueidad de aire	0101	airTightness
Apagado del freno regenerativo	0110	noRegenerativeBrake
Apagado del freno de corriente de Foucault como freno de servicio	0111	noEddyCurrentBrake
Desactivación de las zapatas magnéticas	1000	noMagneticShoeBrake
Apagado de la principal fuente de energía	1001	mainPowerSwitchOff
Apagado del freno de corriente de Foucault como freno de emergencia	1010	noEddyCurrentBrake

Tabla 7: Correlación de tipos de condiciones de vía

En algunos de los tipos de condiciones mostrados en la Tabla 7, se precisa añadir otros valores. Por ejemplo, en el caso de se tenga un área de activación de la bocina, al no estar ese tipo de condición de vía en RailML, se añadirá como "other", es decir que se puede añadir un nuevo tipo para cuadrar con UNISIG, por ejemplo "soundHorn".

En el caso de que se trate de una desactivación del freno de corriente de Foucault, hay que especificar si se desactiva como freno de emergencia o como freno de servicio, característica que se añadiría en RailML en el atributo `description` como "emergency brake" o "service brake" respectivamente.

Por último, si se trata de un túnel, además del elemento `trackCondition`, se añadiría otro elemento de tipo `tunnel`, al que se le añadiría únicamente un identificador y el posicionamiento de área correspondientes.

### 7.5.1.7 Cambios de límites de corriente

La corriente que consume el tren recogida por medio de la catenaria tiene unos límites superiores que evitan dañar tanto la infraestructura como el tren, así como evitan picos de consumos innecesarios.

A la hora de simular las condiciones de vía, es importante saber cuándo cambian estos límites de consumo de corriente varían. En Subset-112 estos cambios en los límites vienen dados por el elemento `currentChange`.

En cambio, en RailML no hay ningún elemento que lo represente, ya que no cuadraría en `electrificationChanges` como los cambios de voltaje (apartado 7.5.1.2), ya que no supone ninguna variación en la electrificación o la catenaria, sino un cambio en los niveles de consumo. Por lo tanto, se podría incluir como una condición de a vía, explicadas en el apartado 7.5.1.6, ya que en versiones anteriores del Subset-112, este parámetro era incluido como una condición de vía.

De este modo, se traduciría a de formato los atributos siguientes:

- Un **identificador**
- **Posicionamiento** puntual
- Para incluir el valor del nuevo **límite de corriente**, habría que añadir un nuevo elemento contenido en este denominado `value`, que especifique este valor nuevo de consumo máximo. Para la traducción habría que multiplicar el valor ofrecido por UNISIG por 10, ya que este cubre los valores de 0 a 1000 A en intervalos de 10 A.
- Aparte de estos atributos, sería conveniente añadir el tipo de `trackCondition` en el atributo `type` como "currentChange".

### ***7.5.1.8 Pasos a nivel***

Un paso a nivel es el cruce entre un tramo de vía ferroviaria y otra vía no ferroviaria como puede ser una carretera o un camino, como indica la Figura 40. Es importante tener en cuenta estas áreas, ya que por ellas circula otro tipo de tráfico que el sistema de control de tráfico ferroviario no puede comprobar, por lo que puede suponer un gran peligro si no se tiene en cuenta y no se prueba el funcionamiento en estas situaciones.

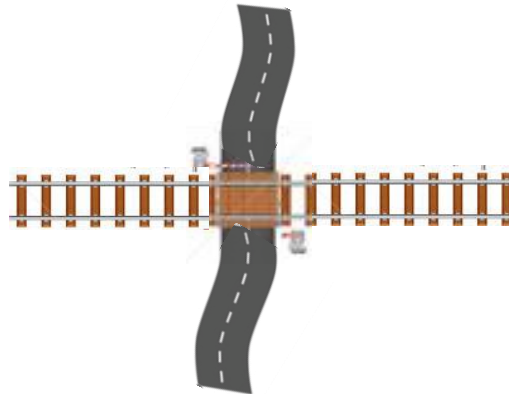


Figura 40: Paso a nivel

Esto se encuentra definido en ambos formatos en el elemento `levelCrossing`. Ambos comparten los atributos:

- Elemento **identificador**
- **Posicionamiento** de área, ya que un paso a nivel puede abarcar varios tramos de vía.

Además de estos atributos, el paso a nivel en UNISIG también incluye:

- `speed`: restricción de velocidad
- `stopArea`: longitud anterior a este área en la que se sitúa la zona de parada.

Como los pasos a nivel de RailML no incluyen estos parámetros, se debe incluir además un elemento de referencia a un cambio de velocidad `speedChange`, como muestra el esquema de la Figura 41. Los valores de la velocidad en UNISIG cubren entre 0 y 600 km/h a intervalos de 5km/h y se incluyen por lo tanto multiplicados por 5 en el atributo `vMax` de `speedChange` en RailML.

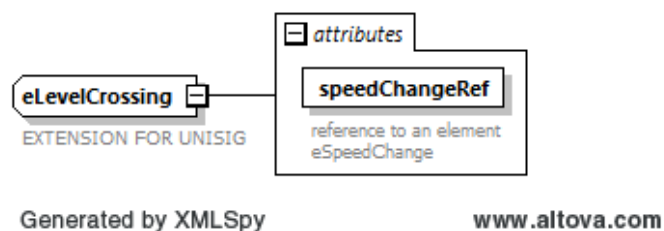


Figura 41: Esquema de la extensión de RailML para `levelCrossing`

En cuanto al posicionamiento de este cambio de velocidad, se situará al comienzo de la zona de parada, es decir, en el punto que se halle restando el comienzo (o los comienzos, en el caso de cubrir varias vías) del paso a nivel, menos la longitud de la zona de parada. Dicha restricción acabará donde termine el paso a nivel, como determina la Figura 42

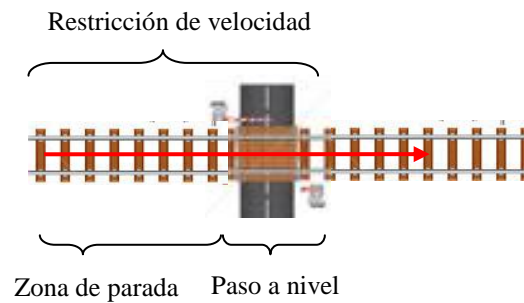


Figura 42: Zonas en un paso a nivel

### 7.5.1.9 Andenes

Otros puntos importantes que tener en cuenta a la hora de describir una infraestructura son las zonas donde se sitúan los andenes o plataformas, es decir, donde el tren puede efectuar una parada para hacer carga y descarga de mercancías, o para que los pasajeros puedan montar o bajarse del tren. Podemos ver un esquema de la plataforma, en este caso situada a ambos lados de la vía, en la Figura 43.

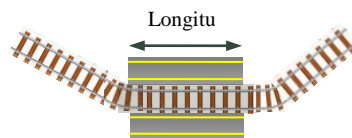


Figura 43: Andén

En el protocolo de UNISIG, los andenes se definen en el elemento `stationPlatform`, mientras que en RailML su elemento correspondiente se denomina `platformEdge`. Ambos comparten los atributos:

- Elemento **identificador**.
- **Posicionamiento** de línea.

En cuanto a los atributos característicos de esta clase de elementos, contamos con:

- **Atura del andén**. Ésta se define como la distancia vertical entre la parte superior de la vía y la parte superior del andén, como se indica en la Figura 44.

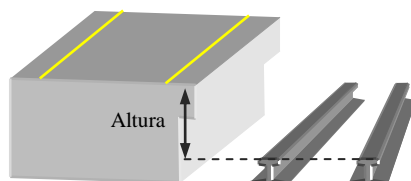


Figura 44: Altura del andén

Esta altura suele venir dada en milímetros, y existen algunos valores normalizados. Es por esto que UNISIG recoge este valor en el atributo `height`, que consta de 4 bits, cuyos valores corresponden a diferentes valores establecidos por las Especificaciones Técnicas de Interoperabilidad, como consta en la Tabla 8.

Valor <code>height</code> UNISIG	Altura de plataforma (mm)
0000	200
0001	300-380
0010	550
0011	580
0100	680
0101	685
0110	730
0111	760
1000	840
1001	900
1010	915
1011	920
1100	960
1101	1100

Tabla 8: Valores para la altura de plataforma en UNISIG [13]

En RailML, se recogen estos valores en el atributo `height`, pero en milímetros, así que se tomarán los valores de la segunda columna de la Tabla 8. En el caso en el que la altura especificada sea 300-380 mm, se pondrá 380 mm para evitar problemas.

- Atributo `side` para ambos formatos. Este atributo define a qué **lado de la vía** se encuentra el andén. En el Subset-112 se define este parámetro con dos bits, que pueden tener los valores especificados en la Tabla 9, tomados para la dirección nominal de la vía.

Valor <code>side</code>	Lado de la vía
UNISIG	
0000	Izquierda
0001	Derecha
0010	Ambos lados

Tabla 9: Valores para el lado del andén en UNISIG [13]

En RailML, por su lado, el atributo `side` es de tipo string y solo admite valores "right" o "left", es decir, solo permite definir el andén a uno de los lados de la vía. En el caso de situarse a ambos lados, se deben crear dos elementos diferentes de tipo `platformEdge`, uno para cada lado de la vía.

- Se añade a este elemento una referencia a una **estación** en el caso de que se conozca a qué estación corresponde, para así agrupar los andenes que pertenezcan a una misma estación. Esta referencia está definida en el atributo `ocpRef`, y hace referencia a un elemento de tipo `ocp`, definido en el apartado 7.5.3.1.

#### 7.5.1.10 Áreas de marcha atrás

En UNISIG se definen ciertas áreas en las que le está permitido al tren dar marcha atrás. Estas áreas se encuadran bajo el elemento `reversingAreas`. Este elemento contiene los atributos siguientes:

- **Identificador.**
- **Posicionamiento** de tipo área.

- **Distancia** que se puede recorrer marcha atrás, definida en 15 bits con un tope de 32670 km, quedando el valor 32767 reservado para una distancia infinita.
- **Velocidad máxima** a la que se puede circular de este modo. Esta velocidad se recoge en 7 bits, es decir en 128 valores, de los cuales se utilizan los 121 primeros (0-120), para registrar velocidades entre 0 y 600 km/h en intervalos de 5km/h [13].

Sin embargo, en RailML no está contemplada la posibilidad de que se pueda ir marcha atrás por lo que no existe un elemento que lo describa. Es por esto que para incluir estos elementos en el caso de ser de necesaria inclusión de un elemento nuevo dentro de `trackElements`, con los mismos atributos que contiene en UNISIG, como muestra el esquema de la Figura 45.

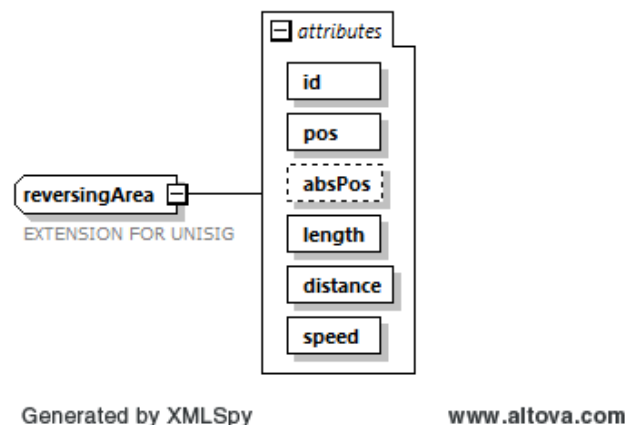


Figura 45: Esquema de la extensión de RailML para reversingArea

## 7.5.2 Sistemas de operación o control

Los elementos de operación o control son aquellos elementos que nos encontramos en la vía y que no son necesariamente tangibles, que definen información necesaria para la administración de la circulación y tráfico ferroviario, como puede ser toda la información intercambiada entre la infraestructura y los trenes informando sobre velocidades, recorridos, horarios, estados de la vía, paradas, etc.

Todos estos elementos se encuadran en UNISIG dentro de los elementos de vía (`trackElementList`), mientras que en RailML encajan dentro de cada tramo de vía, como parte del elemento `ocsElements`.



### 7.5.2.1 Señales

Las señales en la vía corresponden con todas aquellas indicaciones sobre la circulación o tráfico que se encuentran en la vía y/o en las estaciones y no utilizan ningún sistema de telecomunicaciones y que tienen como objeto comunicar información de obligado cumplimiento al tren.

Hay diversos tipos de señales. Según la inmediatez de cumplimiento de la misma, puede ser de cumplimiento inmediato o ser simplemente la repetición o el aviso del contenido de otra señal. Según el contenido pueden ser de limitación de velocidad, de parada, de precaución, de dirección, de paso a nivel, etc. Por otro lado, según el formato pueden ser luminosas, semáforos, señales de pantalla, cartelones, etc.[23]. Se muestran varios ejemplos de señales en la Figura 46.

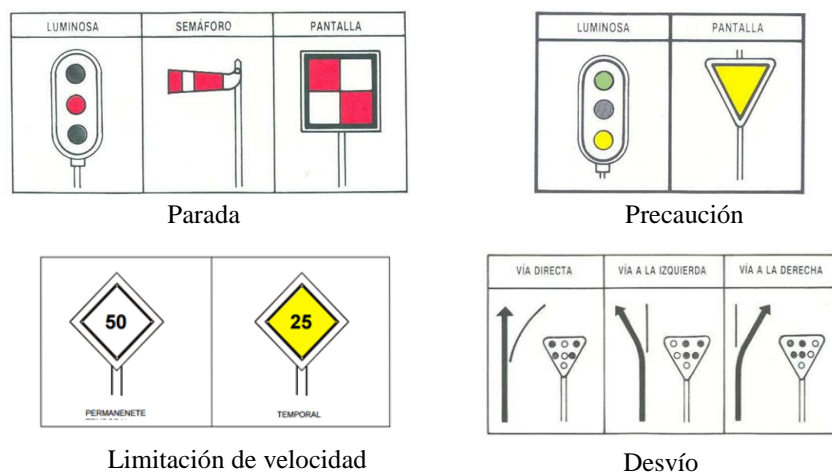


Figura 46: Ejemplos de señales ferroviarias [19]

Estas señales se representan en ambos formatos en el elemento denominado `signal`. En UNISIG no se le atribuye demasiada información aparte de su existencia, ya que el Subset-112 está más centrado en pruebas de interoperabilidad de sistemas de señalización por telecomunicaciones. De este modo incluyen los atributos:

- **Identificador**
- **Posicionamiento** puntual.
- **Descripción** opcional de la señal de tipo string llamado `description`. Esta descripción puede ser incluida en el atributo del mismo nombre en RailML o bien

añadir la información que aporta la descripción en otros atributos de RailML como son la función (*function*), o el tipo (*type*).

En UNISIG, falta información de especial importancia referente al aspecto de la señal, es decir, qué es lo que indica en cada momento.

### 7.5.2.2 Balizas

Una baliza es un transpondedor colocado en la vía. Un transpondedor es un dispositivo que transmite señales a ciertas frecuencias para su recepción por un dispositivo que lee su mensaje.

Para implantación de sistemas ATP/ATC, como por ejemplo, sistemas ERTMS, se emplean balizas que se colocan en la vía, como se aprecia en la Figura 47. Estas balizas contienen unos mensajes que el tren lee a través de un equipo de abordó, recibiendo este así datos sobre los límites de velocidad, órdenes de parada, alertas de peligro o avisos de cualquier otro tipo, y actuando al respecto. Por lo tanto, las balizas suponen parte importante de los métodos de seguridad y protección, de control del tráfico y de la operación ferroviaria de manera automática.



*Figura 47: Baliza ERTMS [8]*

Ya que los protocolos UNISIG 112 y RailML se emplean para ensayos de interoperabilidad de sistemas de señalización en cabina ERTMS, la definición de las balizas que se sitúan en la vía resulta fundamental para la realización de estas pruebas.

Es por este motivo que ambos formatos incluyen el elemento que describe la baliza, denominado en ambos formatos *balise*.

Ambos cuentan con los atributos:

- **Identificador**
- **Posicionamiento** puntual
- **Dirección** estándar.

Aparte de esta información, UNISIG también incluye los datos correspondientes al contenido de la baliza. Esos datos se recogen en una **lista de posibles telegramas** que puede contener la baliza, y aparte otro atributo que contiene una referencia al telegrama que se establece por defecto.

Cada baliza tiene sus datos estructurados como se expone en la Figura 48. De este modo, incluye un conjunto de telegramas que tienen su identificador, una descripción en el atributo `description` del telegrama, y los datos correspondientes.



Figura 48: Estructura elemento baliza

Los datos que contiene la baliza abarcan una gran cantidad de información sobre la infraestructura, que incluye datos sobre puntos importantes de la vía, pasos a nivel, límites de velocidad, distancias a señales o puntos importantes, zonas de peligro, permisos, etc.

Los datos de cada telegrama se guardan de dos formas. La primera es con **los contenidos codificados** en valores hexadecimales en el atributo `codedBaliseData`. Cada uno de los valores está compuesto de dos caracteres y no tienen separación entre ellos. Antes de las pruebas estos datos deben ser verificados, ya que es más difícil identificar un error debido a la codificación.

La otra forma de inclusión de datos es mediante **valores legibles**. Éstos se incluirán en la variable `readableBaliseData`, que contiene una lista de variables con sus diferentes valores.

La inclusión de una lista en lenguaje legible tiene como único objetivo la facilitación de la comprensión de los datos utilizados, ya que comprender los datos que se emplean disponiendo únicamente de código hexadecimal puede resultar entorpecedor y tedioso. Sin embargo, su uso es completamente opcional, ya que las pruebas se pueden llevar a cabo disponiendo únicamente del contenido codificado.

Las variables incluidas en esta lista están definidas por su nombre en `variableName` y su valor codificado en `value`. Cada una de estas variables tiene un formato diferente, y el **significado de cada valor** que se le asignan viene recogido en el Subset 026, que incluye todas las variables que se envían a través de la baliza en el capítulo 7 y el orden y modo de lectura de dichas variables en los mensajes en el capítulo 8.

Para **traducir** todos estos telegramas a RailML, es necesario la creación de nuevos elementos dentro de `balise`, ya que actualmente no recoge ningún campo donde estos encajen. De este modo, se añaden como estados, es decir como elementos tipo `state`, hijos del elemento `balise`. En cada estado se define uno de los telegramas, describiendo con el atributo booleano `disabled`, si están activos o no, es decir, si es el telegrama que se está enviando en el momento o no. En el caso de estar activos, este atributo toma el valor "1" si está desactivado y "0" si está activo.

Además de si está activo o no, es necesario añadir toda la información de cada telegrama. De este modo, se añaden a cada estado los atributos correspondientes al identificador, a los datos codificados y a la descripción del telegrama, los tres de tipo `string`.

Por otra parte, se le añaden a cada estado nuevos subelementos, todos con el mismo formato y cada uno correspondiente con una variable incluida en el telegrama. Cada uno de estos nuevos elementos, que toman el nombre de UNISIG `baliseData`, contiene dos atributos `string`, uno para el nombre de la variable (`variableName`), y otro para su valor (`value`).

Todos estos datos metidos dentro de cada estado como extensión de RailML se muestran en el esquema de la Figura 49.

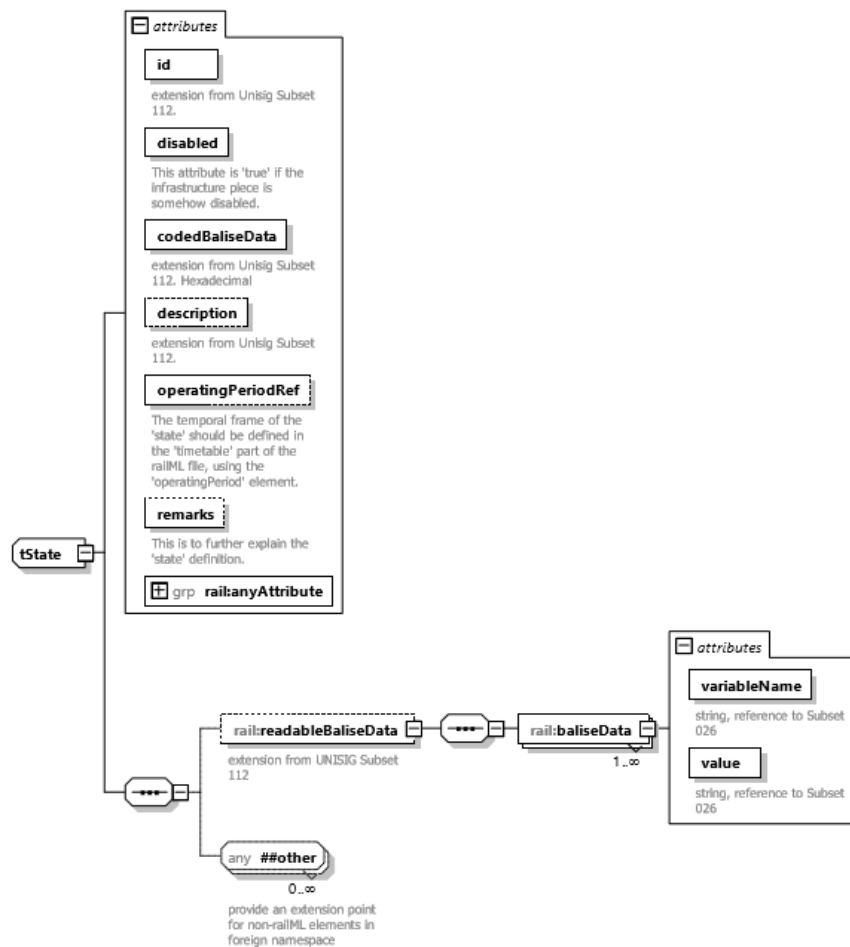


Figura 49: Esquema de la extensión de RailML para state, dentro de balise

### 7.5.2.3 Eurolazos y RIUs

Los Eurolazos y RIUs (Radio Infill Units), son sistemas de señalización ferroviaria que complementan a los sistemas de ETCS de Nivel 1. Ayudan a salvar los inconvenientes que ofrece el transmitir la información desde la vía de manera puntual, transmitiendo parte de ella de forma semicontinua a través de un cable radiante en el caso del Eurolazo y por radio para transmitir un aviso del contenido de la baliza siguiente con el uso de las RIUs.

Estos elementos son cada vez menos utilizados, ya que están siendo sustituidos por otros elementos de radio. De todos modos siguen en uso en ciertas líneas por lo que no se debe despreciar su inclusión en las pruebas de interoperabilidad.

Ambos elementos se encuentran incluidos en la definición de UNISIG, como `euroloop` y `riu` respectivamente. Ambos incluyen un identificador, un posicionamiento de área y una dirección estricta.

En cuanto a su estructura, son muy similares a las de las balizas (apartado 7.5.2.2)., ambos incluyen una lista de mensajes posibles como éstas, como se observa en la Figura 50 y en la Figura 51. En cambio las RIUs incluyen otros atributos propios:

- **countyId**. Identifica el país en el que está situado mediante un entero de 10 bits al que cada número se le es asignado un país por la ERA.
- **riuId**. Identificador del RIU de 14 bits, que el país al que pertenezca se encarga de determinar.
- **riuTelephoneNr**. Número de teléfono de 16 cifras (64 bits) correspondiente a la unidad RIU.
- **networkId**. Identificador de la red a la que se adscribe.



Figura 50: Estructura Eurolazo

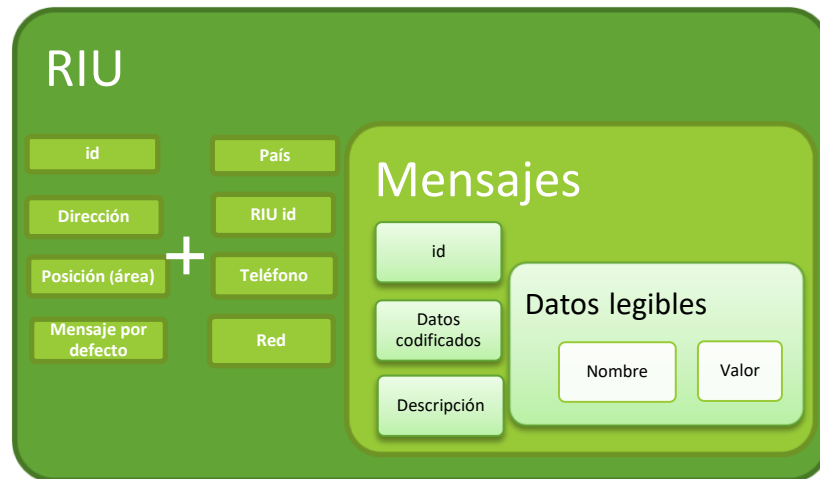


Figura 51: Estructura RIU

En RailML, estos elementos no se contemplan debido a su uso poco extendido actualmente. Sin embargo, deberían añadirse para facilitar así la realización de pruebas sobre líneas con estos sistemas instalados.

Al ser estos sistemas parte de la señalización, y por lo tanto, encargados de la operación y el control de tráfico, van encuadrados como elementos añadidos dentro de `ocsElements` como extensión de RailML, como indican los esquemas de la Figura 52 y de la Figura 53.

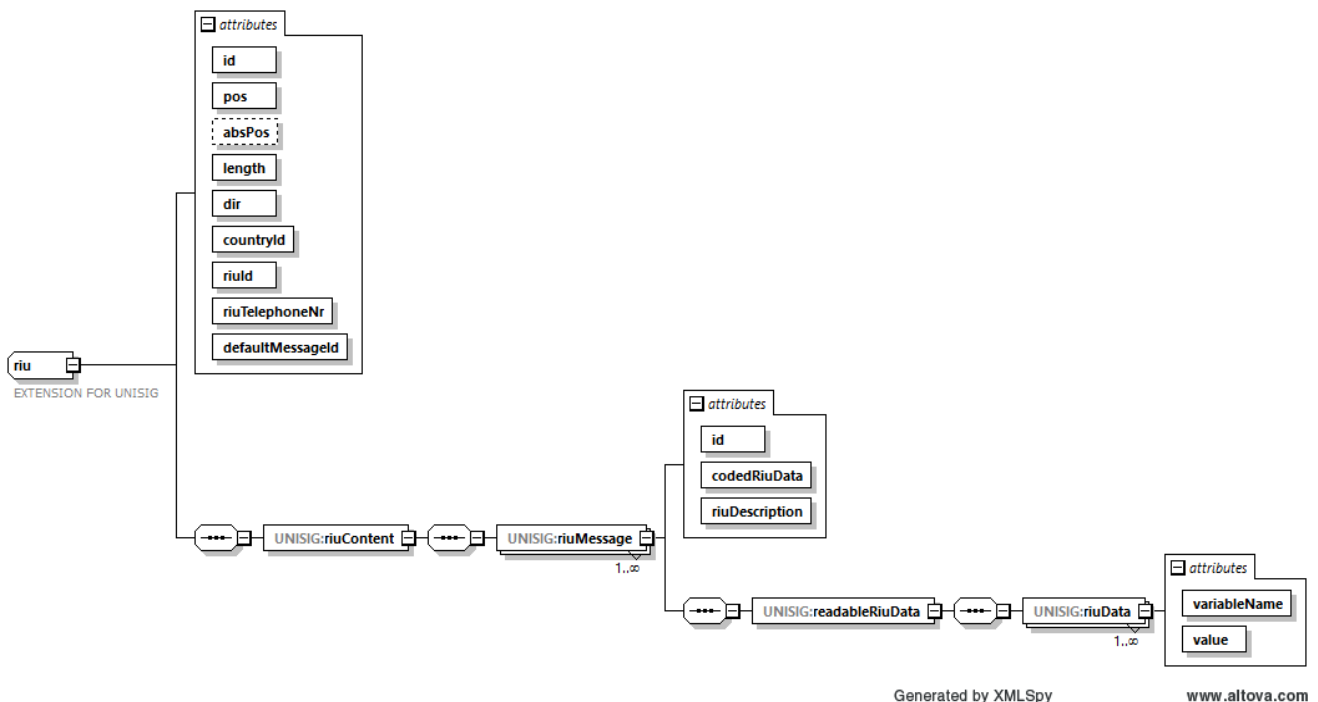


Figura 52: Esquema de la extensión de RailML para riu

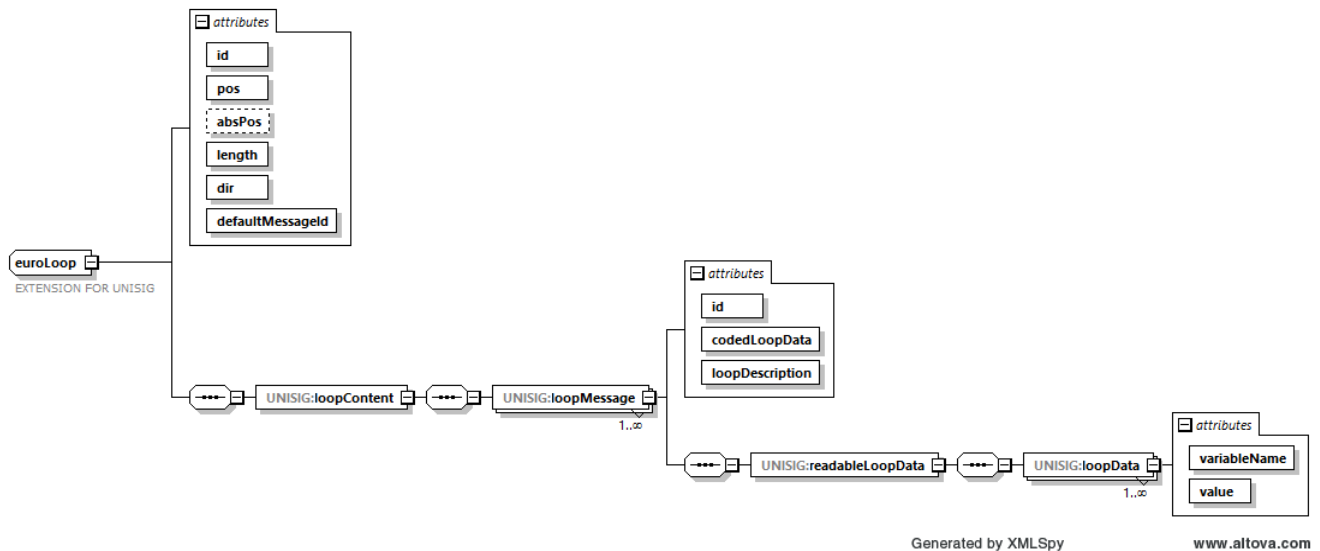


Figura 53: Esquema de la extensión de RailML para euroLoop

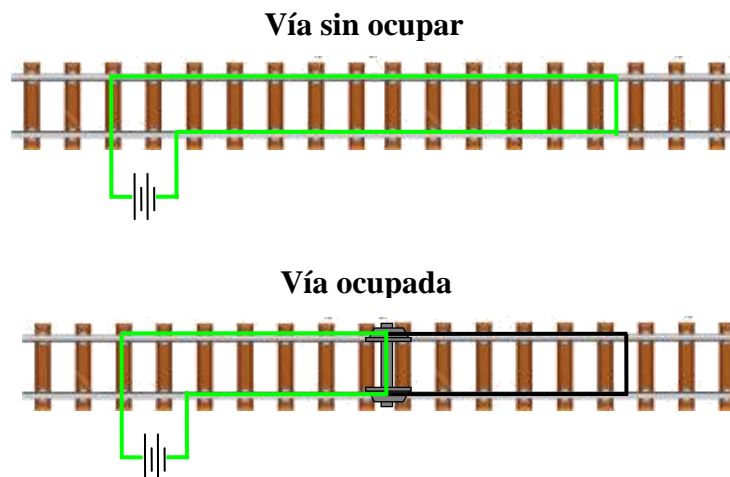
#### 7.5.2.4 Ocupación de la vía

Para el control del tráfico, es fundamental conocer en todo momento dónde se encuentran los trenes, para así determinar en qué momentos dar permiso a otros a efectuar su entrada en ciertos tramos de vía. Sin embargo, es complicado establecer el punto exacto donde se encuentra cada tren si no es mediante geolocalización, en lo que todavía se está trabajando en diversos proyectos.

Es por esto, que lo que se suele hacer es dividir la vía en zonas en las que solo puede haber un tren a la vez. Estas zonas se denominan cantones, y en ellos se sitúan diferentes sistemas de detección de trenes para establecer cuándo hay un tren ocupando un cantón.

Uno de estos sistemas son los **circuitos de vía**. Éstos se consiguen suministrando un voltaje a ambos carriles, generando un circuito cerrado que se cortocircuita cuando pasa el tren y sus ruedas unen ambos carriles. De este modo, si no se detectan los niveles de voltaje esperados en el borde final del circuito se determina que ese tramo de vía se encuentra ocupado, como indica la Figura 54.



*Figura 54: Circuito de vía*

A parte de los circuitos de vía, se pueden emplear también otros tipos de detectores compuestos por sensores de diversas índoles, como pueden ser mecánicos, ópticos, inductivos, etc. Estos sensores se encargan de detectar partes del tren como los ejes.

Para definir los cantones, UNISIG emplea el elemento `trackOccupation`. Este solo está definido por un identificador y por un posicionamiento de área que corresponde con la zona en la que se puede detectar la presencia de un tren.

Por otra parte, en RailML existen definidos elementos de detección de trenes, divididos en **elementos de detección** y en **circuitos de vía**. Como `trackOccupation` hace referencia a un área y no a un punto en concreto como haría referencia los dispositivos puntuales de detección, es más conveniente encuadrarlo como circuito de vía.

En cambio, en el elemento que corresponde al circuito de vía solo aparece el borde del circuito que queda aislado cuando pasa un tren. Este borde aislado debería situarse al principio del mismo para poder establecer las señales pertinentes para impedir la entrada de otro tren cuando ese cantón esté ocupado. Por lo tanto, en el elemento `trackCircuitBorder` de RailML, se le incluye la posición correspondiente al comienzo del área definida en cada elemento `trackOccupation` de UNISIG, además de su identificador.

### 7.5.2.5 Sistemas NTC

Los sistemas NTC, son aquellos que siguen un sistema ATP nacional, por lo que se precisa para su uso definir dicho sistema.

En los ensayos de interoperabilidad en los que se emplea el protocolo de UNISIG se contempla la inclusión de los elementos correspondientes de vía del nivel NTC, aunque son de definición opcional, ya que las partes de más interés de estos ensayos es la interoperabilidad entre sistemas ERTMS, no con los sistemas nacionales. Es por esto que los elementos correspondientes al nivel NTC son poco utilizados.

En el Subset-112 se incluyen los elementos del lado de vía del nivel NTC en dos elementos, `ntcDevice` y `ntcLoop`, que corresponden a la definición de dispositivos como balizas y cable radiante de un sistema nacional respectivamente.

Estos elementos disponen en UNISIG de una estructura similar a la de sistemas anteriores (apartados 7.5.2.2 y 7.5.2.3). Los dispositivos NTC tienen una estructura similar a las balizas (Figura 55) y los lazos NTC a los Eurolazos (Figura 56), pero añadiendo en ambos el tipo de NTC usado, en `ntcType`, de tipo string.



Figura 55: Estructura Dispositivo NTC



Figura 56: Estructura Lazo NTC

Las variables enviadas corresponden con las enviadas por la baliza, recogidas en el Subset 026 en los capítulos 7 y 8 para la determinación de las variables y el formato de los mensajes respectivamente.

En RailML no aparecen incluidos estos elementos ni ninguno con correspondencia directa, ya que estos elementos no aportan información demasiado relevante en estas pruebas, al tener que ver con sistemas nacionales y no con ERTMS, que son para los que se realizan estas pruebas.

De este modo, en el caso de que fuera necesaria su inclusión para determinadas pruebas, se debería incluir en RailML como extensión, como elementos nuevos dentro de `ocsElements`, ya que se tratan de elementos correspondientes a un sistema de señalización, y por lo tanto de un sistema de control y de operación. Estos elementos contarían con la misma estructura y formato que tienen en Subset-112, como muestran los esquemas de las extensiones correspondientes de RailML que aparecen en la Figura 57 y en la Figura 58.

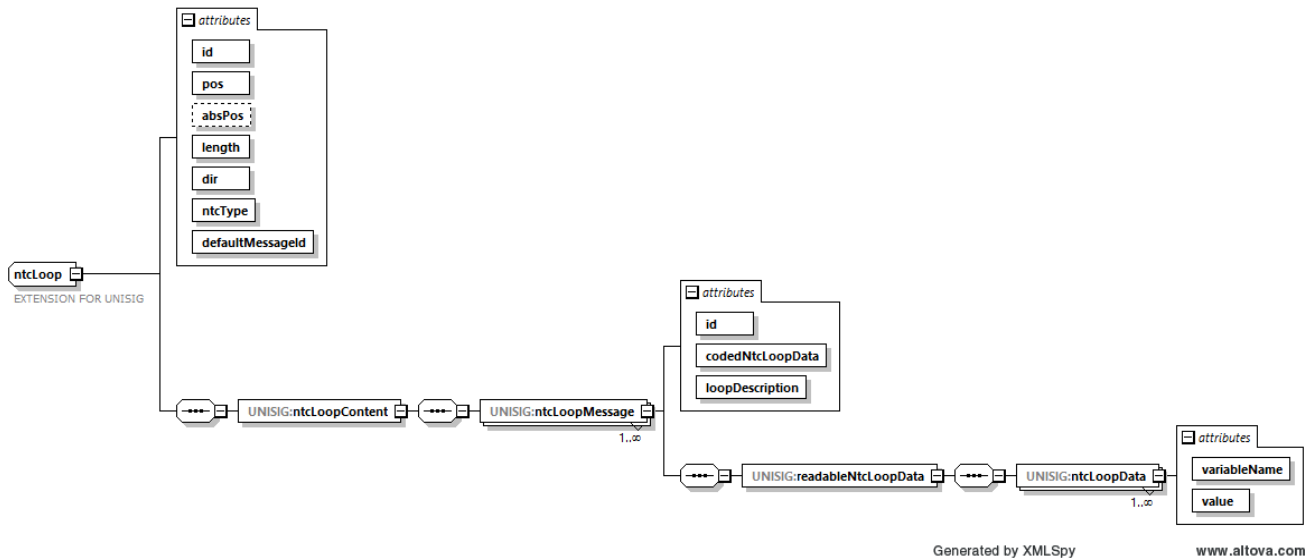


Figura 57: Esquema de la extensión de RailML para `ntcLoop`

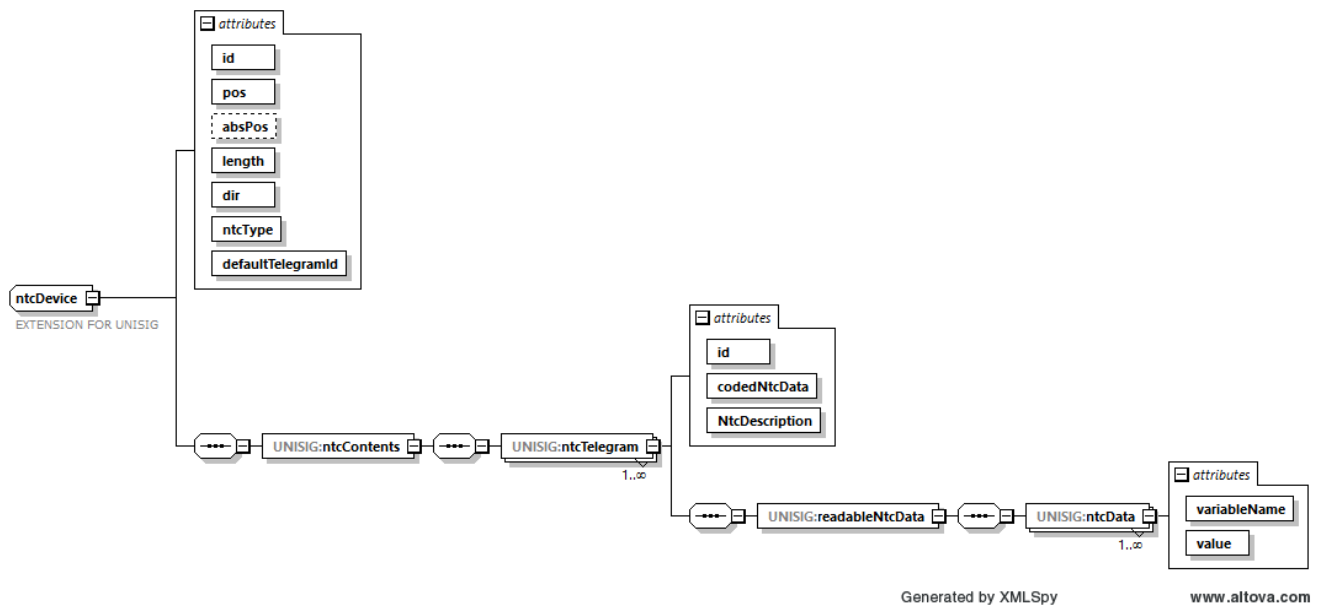


Figura 58: Esquema de la extensión de RailML para `ntcLoop`

### 7.5.3 Puntos de operación o control

Los puntos de control operacional son aquellas zonas físicas que facilitan la operación o control de trenes, es decir, son puntos concretos en la vía que permiten el cambio de trenes, cambio de vías o de cargas. Estos puntos quedan encuadrados en RailML dentro

como elementos de tipo `ocp`, fuera de la definición de cada tramo de vía, aunque quedan referenciados dentro de ellos.

### 7.5.3.1 Estaciones

Las estaciones son agrupaciones de andenes donde los trenes pueden efectuar una parada para que así los pasajeros puedan bajar o subir al tren.

En UNISIG las estaciones no están definidas. Sin embargo, sí que tiene en cuenta los andenes. Por lo tanto, si se conocen los andenes que corresponden a la misma estación, o un andén es dividido en dos elementos diferentes por tener plataforma a ambos lados de la vía (como se indica en el apartado 7.5.1.9), se puede establecer en RailML un elemento al que estos andenes hagan referencia y represente una estación.

Una estación supone un elemento de tipo `ocp` con un atributo identificador y un elemento hijo de tipo `propOperational` que cuenta con dos atributos:

- **operationalType**: tipo de `ocp`, que adquiere el valor "station" para definir que se trata de una estación.
- **trafficType**: tipo de tráfico, con valor "passengers", ya que en una estación el tráfico es únicamente de pasajeros.

### 7.5.3.2 Áreas de maniobras

Las áreas de maniobras son aquellas zonas en las que se efectúa el cambio del tren de una vía a otra, o en la que se cambia el orden de los vagones o se sustituye o se elimina un vagón por otro. También se suelen emplear para poner el tren a circular en sentido contrario, cambiando la cabeza tractora (locomotora) de lado de la cadena de vagones, mediante la acción de una locomotora de maniobras. Se muestra un ejemplo de este tipo de zonas en la Figura 59.



Figura 59: Área de Maniobras de Barcelona-Can Tunis[24]

Al tratarse de una zona donde se facilita la operación y control de los trenes que circulan por cada vía y de su composición, se puede considerar un punto de control u operación, por lo que en RailML encaja dentro del elemento `ocp`. Este cuenta un elemento hijo de tipo `propOperational` que cuenta con un atributo para definir **tipo de tráfico** (`trafficType`), con valor "shunting", ya que se trata de un área de maniobras.

En UNISIG este elemento aparece encuadrado en el elemento `shuntingArea`. Para la traducción se tienen en cuenta los atributos:

- **Identificador**
- Este elemento tiene en UNISIG una lista de referencias a **balizas** que actúan en esta zona, las cuales se añadirán como un nuevo elemento de `ocp`, con la misma estructura que en UNISIG. De este modo, el elemento `ocp` incluye el identificador de las balizas que corresponda, mediante una referencia a un grupo de balizas, que se añade en RailML como extensión, como indica la Figura 60.

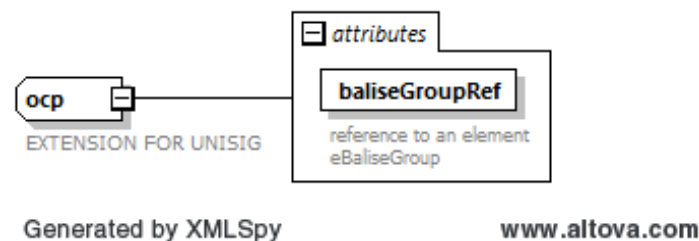


Figura 60: Extensión de RailML para `ocp`

En cuanto al posicionamiento, este elemento de RailML no va dentro de cada tramo de vía, por lo que se ha de referenciar dentro de ella. Para ello en cada tramo que forme

parte de este área se añade en topología el elemento `crossSection`, que funciona como referencia a un punto de operación.

Por lo tanto, dentro de este elemento `crossSection` se ubicará la referencia al identificador del elemento `ocp` dentro del atributo `ocpIDRef`, así como la posición del tramo donde comienza la zona de maniobras.

#### 7.5.4 Atributos de la infraestructura

Algunas de las características de ciertos áreas de la vía suelen estar estandarizadas, como pueden ser gálilos, electrificaciones, límites de velocidades, etc. Estos valores estandarizados sólo pueden tomar ciertos valores, ya sea por simplificación, por normas de cada país, regulaciones europeas, o motivos de diversa índole.

Es por esto que RailML dispone dentro de su infraestructura un elemento para agrupar diversos tipos de atributos con estándares. Este elemento se denomina `infraAttributes`, y cada uno de ellos identifica un grupo de atributos que se aplica a la vez, como puede suceder por ejemplo, cuando se cruza una frontera y se han de asumir ciertos límite en función de las regulaciones de dicho país.

Dentro de `infraAttributes` hay definidos varios tipos de valores, además de uno adicional llamado `generalnfraAttributes`, para encuadrar todos aquellos atributos que no encajen en los ya ofrecidos. Cada uno de los atributos de la infraestructura aquí definidos contiene los atributos para su nombre y su valor, ambos de tipo string.

Cada uno de estos grupos de atributos de infraestructura han de ser referenciados con su id desde cada tramo de vía `track` que se encuentre afectado por esos valores, mediante el elemento `infraAttrGroupRef`.

##### 7.5.4.1 Valores nacionales

Los valores nacionales corresponden con una serie de restricciones y condiciones impuestos por las normativas de cada país, que cambian al cruzar las fronteras.



UNISIG recoge estos valores dentro del elemento `nationalValueChange`, mientras que en RailML, al tratarse de valores estándares que afectan a cada país, se define como grupo de atributos de infraestructura en `infraAttributes`.

En UNISIG la posición es puntual y se establece una dirección en la que es válido, ya que define el punto de cambio de todos estos valores por traspasar una frontera. En RailML en cambio, se referencia el grupo de atributos desde cada tramo de vía afectado, sin necesidad por lo tanto de especificar la dirección.

Ninguno de los atributos de infraestructura posibles ofrecidos en RailML corresponde con los de UNISIG, por lo que se incluirán todos ellos como `generalInfraAttributes`, empleando las mismas estructuras que se emplean en las especificaciones de UNISIG, pero incluyéndose como extensión de RailML, como se indica en el esquema de la Figura 61.



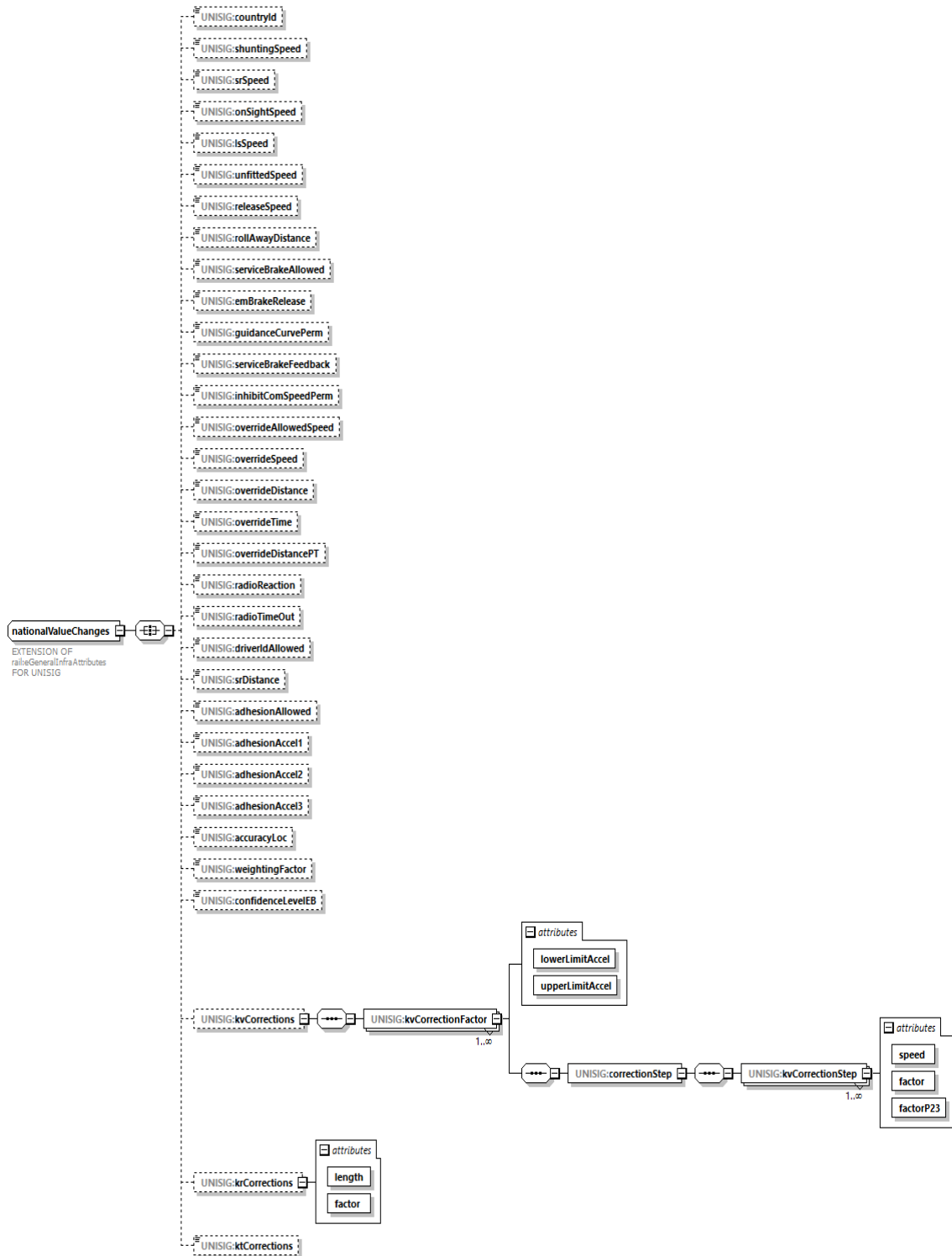


Figura 61: Esquema de la extensión de RailML para nationalValueChanges

Entre estos valores todas las velocidades se especifican con un entero de 7 bits que representa las velocidades entre 0 y 600 km/h a intervalos de 5 km/h. Las aceleraciones utilizan 6 bits para definir valores entre 0 y 3,15 m/s<sup>2</sup> a intervalos de 0,05 m/s<sup>2</sup>. Las

distancias se definen en 15 bits tomando valores entre 0 y 327,660 km, con intervalos de 10 cm, 1m o 10 m según se indique.

Entre los valores nacionales se incluyen velocidades máximas para cada modo operativo, especificados en la Tabla 10 (Los modos operativos vienen especificados más adelante en la Tabla 17).

Atributo	Modo operativo
shuntingSpeed	Shunting
srSpeed	Staff Responsible
onSightSpeed	On sight
lsSpeed	Limited Supervision.
unfittedSpeed	Unfitted

Tabla 10: Velocidades máximas de Valores nacionales

También se incluyen otros parámetros que no tienen que ver con estas velocidades.

- **releaseSpeed:** velocidad máxima a la que se puede aproximar el maquinista bajo su responsabilidad a un punto de peligro tras recibir la señal correspondiente. Esto se puede hacer por ejemplo, para aproximarse a la siguiente baliza sin sobrepasar el punto de peligro
- **rollAwayDistance:** distancia máxima de deslizamiento antes de activar los frenos de emergencia.
- **serviceBrakeAllowed:** indica si el freno de servicio está permitido
- **emBrakerelease:** permiso para desactivar el freno de emergencia si se ha reducido la velocidad por debajo de la permitida (valor "1" del atributo) o se está en parada (valor "0" de la variable)
- **guidanceCurvePerm:** permiso para utilizar la curva de guía o no ("1" si está permitido, "0" si no)
- **serviceBrakeFeedback:** permiso para utilizar el feedback del freno de servicio ("1" si está permitido, "0" si no)
- **inhibitComSpeedPerm:** permiso para inhibir la compensación de las imprecisiones de las medidas de velocidad ("1" si está permitido, "0" si no)

- **overrideAllowedSpeed**: límite de velocidad para permitir al maquinista seleccionar el modo "override" y así poder pasar una señal de parada por alguna emergencia.
- **overrideSpeed**: límite de velocidad para el modo "override"
- **overrideDistance**: distancia máxima recorrido en el modo "override"
- **overrideTime**: tiempo máximo en el modo "override" ( entre 0 y 255 s)
- **overrideDistancePT**: distancia máxima a recorrer marcha atrás en el modo "override"
- **radioReaction**: reacción del tren cuando se sobrepasa el tiempo máximo sin recibir la señal de que todo está en orden y bajo seguridad, con los valores mostrados en la Tabla 11.

radioReaction	Significado
00	Entrar en modo "Trip"
01	Aplicar el freno de servicio
10	No reaccionar

Tabla 11: Valores de radioReaction

- **radioTimeOut**: el tiempo máximo sin recibir la señal de que todo está en orden y bajo seguridad, entre 0 y 254 segundos, siendo el valor 255 correspondiente a tiempo infinito.
- **driverIdAllowed**: permiso para introducir el identificador del maquinista ("1" si está permitido, "0" si no).
- **srDistance**: máxima distancia a recorrer en el modo "Staff Responsible"
- **adhesionAllowed**: permiso que el maquinista modifique el valor del factor de adhesión ("1" si está permitido, "0" si no).
- **adhesionAccel1**: máxima deceleración bajo condiciones de adhesión reducidas para trenes de pasajeros con freno en la posición "P" y sin frenos especiales que no dependan de la adhesión.
- **adhesionAccel2**: máxima deceleración bajo condiciones de adhesión reducidas para trenes de pasajeros con freno en la posición "P" y con frenos especiales que no dependan de la adhesión.

- **adhesionAccel13**: máxima deceleración bajo condiciones de adhesión reducidas para trenes de mercancías con freno en la posición "P" o "G".
- **accuracyLoc**: precisión de localización de las balizas, tomando valores enteros en 6 bits entre 0 y 63 m.
- **weightingFactor**: factor para calcular la adhesión, tomando valores enteros en 6 bits entre 0 y 1 a intervalos de 0,05.
- **confidenceLevelEB**: nivel de fiabilidad de los frenos de emergencia, usando 4 bits según muestra la Tabla 12.

<b>confidenceLevelEB</b>	<b>Valor</b>
0	50 %
1	90 %
2	99 %
3	99,9 %
4	99,99 %
5	99,999 %
6	99,9999 %
7	99,99999 %
8	99,999999 %
9	99,9999999 %

Tabla 12: Valores de *confidenceLevelEB*

- **kvCorrections**: contienen una lista de diferentes factores de corrección Kv. Estos se clasifican primero según los límites de deceleración bajo adhesión reducida. Para cada pareja de límites de deceleración se establece una lista de valores para el factor Kv en función de la velocidad. Los factores Kv toman valores enteros en 7 bits que corresponden con valores entre 0 y 2,54 en intervalos de 0,02.
- **krCorrections**: contienen una lista de diferentes factores de corrección Kr, tomando valores enteros que corresponden con valores en 5 bits entre 0 y 1,55 en intervalos de 0,05, según la longitud del tren, definida por el parámetro *length*, cuyos valores se muestran en la Tabla 13.

length	Valor
0	0
1	25
2	50
3	75
4	100
5	150
6	200
7	300
...	Salto de 100 en 100
3	300

Tabla 13: Valores de length (krCorrections)

- **ktCorrections**: factor de corrección integrado Kt. Toma valores enteros en 5 bits que corresponden con valores entre 0 y 1,55 en intervalos de 0,05.

#### 7.5.4.2 Áreas de ETCS

Un área de ETCS se refiere a una zona con sistemas ETCS implantados que pueden ser de varios niveles, pero con propiedades estáticas definidas en todo el área. Al tener las mismas propiedades en todo el área se pueden definir como atributo de vía.

Las áreas ETCS (etcsArea en Subset-112) contienen los atributos:

- **Identificador**
- **Posicionamiento** de tipo área.
- Todos los **niveles de ETCS** que se aplican a esta área en orden de prioridad, establecidos en etcsLevels. Cada uno de los niveles aquí incluidos tiene un atributo para indicar el nivel al que representa mediante números enteros como se indica en la Tabla 14.

etcsLevel	Nivel
0	Nivel 0
1	Nivel NTC
2	Nivel 1
3	Nivel 2
4	Nivel 3

Tabla 14: Valores de etcsLevel

Por otro lado cada nivel también incluye el valor `ntcId`, compuesto de 8 bits de los que a cada valor se le adjudica un sistema NTC diferente según el país o la región.

- Identificador del **país** donde se aplica (`countryId`)
- **Identificador RBC** (`rbcId`)
- **Número de teléfono** de 16 cifras del RBC (`rbcTelephoneNr`)
- Identificador de la **red** (`networkId`),
- Atributo que define si se debe poner en marcha el correspondiente equipo ETCS o continuar inactivo (`sleeping`) y la versión de ETCS utilizada (`etcsVersion`), de 7 bits correspondientes con los valores mostrados en la Tabla 15.

etcsVersion	Versión
000 XXXX	Versiones anteriores
001 0000	Versión 1.0
001 0001	Versión 1.1
001 0010 - 001 1111	Valores no válidos
010 0000	Versión 2.0
010 0001 - 111 1111	Reservados para futuras versiones

Tabla 15: Valores etcsVersion

En RailML, se establecerían todos estos atributos como un atributo general `generalInfraAttribute`, como muestra el esquema de la Figura 62.

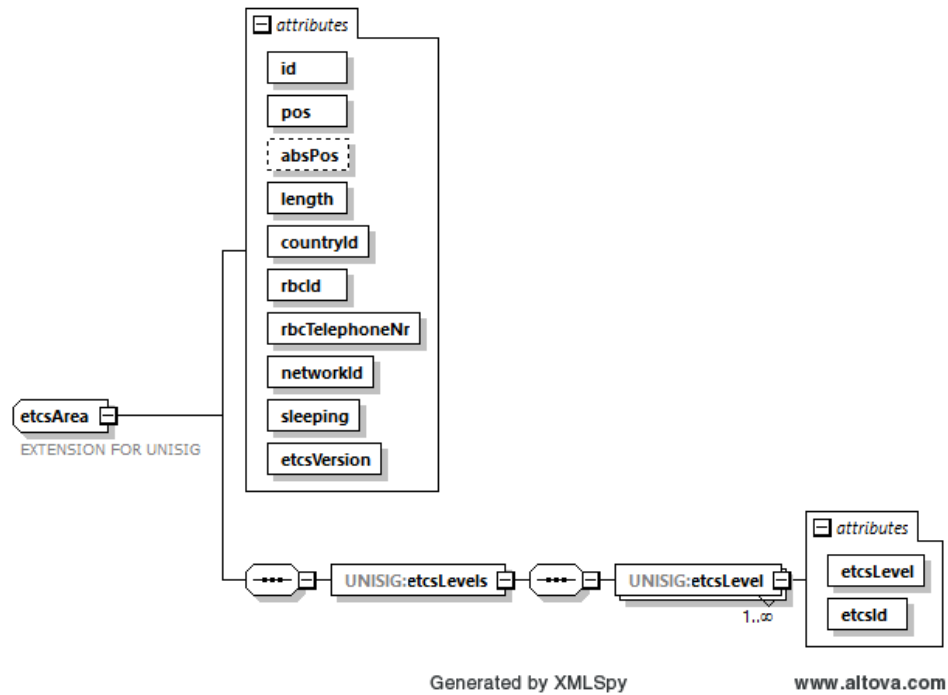


Figura 62: Esquema de extensión de RailML para etcsArea

Además, en el caso de incluir Niveles del 2 en adelante, se le añadiría a este grupo el atributo de infraestructura `trainRadio`, especificando el **tipo de Radio** empleado en su atributo `radioSystem`, que tomaría el valor "GSM-R".

Para marcar el **posicionamiento** se referencia este grupo de atributos en cada tramo de vía afectado.

## 7.6 PROPIEDADES DE LA INFRAESTRUCTURA

En el protocolo UNISIG se incluye un elemento denominado `trackPropertiesType`. Este elemento incluye información acerca de los proveedores de la infraestructura. Es poco utilizado ya que sean cuales sean los administradores de la infraestructura los resultados de las pruebas no varían.

La mayoría de estos datos encajan dentro de los atributos de `infrastructure`. De este modo se traducen los siguientes atributos de UNISIG:

- **description:** es un atributo opcional que incluye una descripción de la infraestructura. Se traduce dentro del atributo `description` en RailML.
- **administration:** corresponde con el nombre del dueño de la infraestructura. Existe un elemento para este dato dentro de `infraAttributes`, pero precisa referenciarlo en cada uno de los tramos. Por lo tanto, para un dato como este que afecta a toda la infraestructura, conviene más añadirlo a su descripción en `description`
- **trackName:** corresponde con el nombre de la infraestructura, por ejemplo "Línea de Alta Velocidad Madrid-Valencia". Se incluye en el atributo `name`.
- **trackVersion:** corresponde con la versión de la vía en términos ingenieriles. Se incluye dentro de la descripción.
- **RBCsupplier:** define al proveedor de RBCs, en el caso de que en algún punto de la línea se utilice ERTMS de nivel 2. Se incluye dentro de la descripción
- **RBCversion:** define la versión de RBC y se incluye dentro de la descripción.
- **TRKsupplier:** corresponde al proveedor de dispositivos de ERTMS como LEUs. Se incluye dentro de la descripción.
- **TRKversion:** corresponde con la versión de dichos dispositivos .Se incluye dentro de la descripción.

El único dato que no encaja como atributo de la infraestructura es la fecha en la que se genera la documentación con los datos para los ensayos (`date`), ya que no es una característica de la infraestructura sino del documento. Es por esto que se incluye dentro de los metadatos del elemento global `railml`.



## 7.7 RUTAS

Las rutas son los posibles caminos que puede tomar el tren a lo largo de la infraestructura, ya que no todas las combinaciones tienen por qué ser posibles debido a limitaciones de operación.

En UNISIG estas listas de segmentos conectados, están recogidas en el elemento `routeList`. En esta lista de rutas vienen recogidas todas las rutas posibles, y no es posible emplear una ruta que no se encuentre predefinida dentro de esta lista.

Esta lista recoge múltiples rutas de la clase `routeType`. Este elemento dispone de varios atributos:

- Un identificador `id` de tipo `string`.
- Una lista llamada `routeSegments` que contiene elementos denominados `segmentID`. Estos elementos contienen los identificadores de cada segmento que compone la ruta.
- Una lista llamada `routeDevices` que contiene elementos del tipo `routeDevice`, cada uno correspondiente con un dispositivo de señalización que afecte a la ruta. Cada uno de estos elementos contiene varios atributos. Uno de ellos es el tipo de dispositivo del que se trata, recogido en `deviceType`, de tipo entero, en el que cada valor numérico corresponde con un tipo como se muestra en la Tabla 16.

<code>deviceType</code>	Significado
1	Eurobaliza
2	Eurolazo
3	RIU
4	Dispositivo NTC
5	Lazo NTC

Tabla 16: Valores `deviceType`

Además incluye el identificador del dispositivo correspondiente, y el del mensaje válido para esta ruta de todos los que contenga el dispositivo.

- Una lista llamada `routeSignals` que contiene elementos del tipo `routeSignal`, cada uno correspondiente con una señal que afecta a la ruta. Cada uno de estos elementos contiene dos atributos. Uno de ellos es el identificador de la señal a la que hace referencia, y el otro es un atributo opcional llamado `signalAspect`, de tipo `string`, que describe el aspecto de la señal. El aspecto de la señal es por ejemplo, si una señal está en rojo o en verde.
- Una lista llamada `routeModes` que contiene elementos del tipo `routeMode`, cada uno correspondiente con los tipos básicos de operación que se establecen a bordo en dicha ruta. Cada modo contiene un atributo denominado `mode`, que puede adquirir diferentes valores numéricos según el modo operativo [13], así como la descripción de cada uno de ellos [25], como se establece en la Tabla 17.

<b>mode</b>	<b>Significado</b>	<b>Descripción</b>
0	Full Supervision	Supervisión completa por ERTMS
1	On sight	Entrada a una vía ocupada, a una velocidad que permita parar ante obstáculos
2	Staff Responsible	Responsabilidad completa sobre el maquinista
3	Shunting	Maniobras
4	Unfitted	La línea no está equipada con ERTMS
5	Sleeping	Cuando se tienen varios equipos de ERTMS a bordo y solo se permite el uso de uno de ellos, el otro está en este modo.
6	Stand By	Modo por defecto
7	Trip	Un permiso de movimiento ha sido sobrepasado, hasta que el maquinista se percata
8	Post Trip	Cuando se está en modo Trip y el maquinista se percata. Se activan los frenos de emergencia
9	System Failure	Fallo de ERTMS. Se requieren frenos de emergencia
10	Isolation	El maquinista aísla el sistema ERTMS ante un fallo del mismo
11	Non Leading	Modo de la locomotora trasera cuando se circula en tándem (varias locomotoras)
12	Limited Supervision	Supervisión ERTMS limitada
13	National System	Sistema de señalización nacional (NTC)
14	Reversing	Se permite marcha atrás
15	Passive Shunting	Maniobras ayudado por una locomotora de maniobras

Tabla 17: Modos operativos



- Una descripción opcional de la ruta, con un elemento llamado `routeDescription`.
- En el caso de que esta ruta tenga uno de sus extremos en una ruta o vía externa, se puede añadir información adicional acerca de dicha ruta mediante los elementos `externalRouteID` y `externalTrackside`, para referenciar los identificadores de la ruta externa o la vía externa, respectivamente.

En RailML no están definidas aún las rutas. Sin embargo, se está desarrollando actualmente un elemento denominado `interlocking`, que pretende recoger todos los datos sobre las rutas que se manejan actualmente por los diferentes administradores de infraestructura y que se recogen en tablas de ruta. No existe actualmente información oficial sobre este elemento, únicamente prototipos recogidos en las propuestas para el desarrollo de la siguiente versión de RailML, RailML 3.0.

## 7.8 VISUALIZACIÓN

A la hora de realizar ensayos de interoperabilidad, resulta de gran utilidad disponer de una interfaz que permita una visualización sencilla del desarrollo de la prueba, que permita a su vez desarrollar conclusiones y realizar cambios y mejoras de forma rápida e intuitiva. En la Figura 63 se puede observar un ejemplo de la visualización de un tramo de vía en el que se encuentra una estación.



*Figura 63: Visualización de vía*

Para esto resulta extremadamente útil la visualización esquemática de la topología de vía, mostrando donde se encuentran puntos claves de la misma, así como la localización en la misma de todos los dispositivos de señalización, o en qué punto se encuentran los trenes en cada momento.

En el protocolo de UNISIG se ofrecen las coordenadas cartesianas de todas las conexiones, así como los puntos en los que en cada tramo se observa un giro (bends, en el elemento segment). Estas coordenadas y posiciones no hacen referencia a ningún punto real de la topología, sus únicos fines son de dibujo del esquema de la infraestructura para facilitar la visualización.

Sin embargo, en RailML esto supone una carencia importante. En la versión anterior, RailML 2.1, se eliminó un elemento recogido en versiones anteriores denominado `infrastructureVisualizations`, que incluía toda la información necesaria de todos los elementos a representar, con sus correspondientes coordenadas cartesianas para su dibujo, y las referencias a los identificadores de cada elemento. Desde la eliminación de este elemento no existe en este formato el modo de registrar los datos correspondientes a la representación gráfica de la infraestructura.

## 8 PRESUPUESTO DEL PROYECTO

Para estimar el coste del proyecto se han tenido en cuenta varios factores.

Primero, se han considerado los materiales empleados. En este caso los materiales necesarios suponen un ordenador del que se provee al alumno en prácticas y el software empleado para trabajar con los esquemas de RailML, Altova XML Spy.

Por otra parte, las horas de ingeniería empleadas suponen el coste mayor. Estas incluyen las horas de estudio previo y preparación, que son bastantes ya que al entrar a las prácticas no se disponía de ningún conocimiento acerca del ámbito ferroviario. Además, se incluyen las horas dedicadas al análisis comparativo realizado, y al dedicado al trabajo final. El coste de la hora de ingeniería corresponde con el precio que ADIF pone a los proyectos realizados por ingenieros junior.

Todos los costes del proyecto se incluyen en la Tabla 18.

Coste de materiales	Unidades	Precio (€)	Coste(€)
Ordenador	1	500	500
Software	1	399	399
Total:			899

Coste de ingeniería	Horas	Precio (€)	Coste(€)
Estudio previo y preparación	30	28	840
Análisis comparativo	120	28	3360
Trabajo realizado	80	28	2240
Total:			6440

Coste total del proyecto	7339 €
--------------------------	--------

Tabla 18: Costes

## 9 CONCLUSIONES Y PROPUESTAS

---

Tras el trabajo realizado, con el consecuente estudio de ambos formatos de datos para aplicaciones ferroviarias, se han llegado a conclusiones acerca de la utilidad de ciertos elementos en ambos formatos, así como la necesidad de incorporación de elementos nuevos o el planteamiento de la eliminación de otros.

### 9.1 PROPUESTAS DE MEJORA

#### 9.1.1 UNISIG SS 112

El formato UNISIG SS 112 tiene varias carencias que impiden que sea un formato idóneo para proyectos como el VITE, razones por las que este formato se debería sustituir por otro como RailML o solucionar dichos problemas.

- 1) UNISIG, a diferencia de RailML, sólo incluye datos acerca de la infraestructura, sin cubrir otros datos sobre el **material rodante** ni sobre los **horarios**, por lo que resulta un formato menos completo.
- 2) Los elementos topológicos de UNISIG pueden conducir a errores y malinterpretaciones acerca del **posicionamiento**, ya que incluyen dos tipos de coordenadas, las cartesianas para fines de visualización y los puntos kilométricos. Esto se podría solucionar tratando las visualizaciones por separado en diferentes elementos.
- 3) La falta de elementos topológicos de cuarto grado impide la definición concisa de **cruces y travesías**, obligando a que estas sean definidas mediante la unión de dos puntos, lo cual resulta demasiado complicado y confuso. Además existe una gran ambigüedad y falta de información a la hora de definir cruzamientos, travesías de unión simple y travesías de unión doble, ya que las tres se definen del mismo modo, por lo que no existe ninguna clase de diferenciación entre ellas. Estos problemas se solucionarían añadiendo un elemento topológico que se uniese a cuatro conexiones, y que incluyese información acerca de qué tipo de cruce se trata.
- 4) Existen algunos elementos en los que algunos valores incluyen varias posibilidades, induciendo cierta **ambigüedad**. Esto sucede en los cambios en la electrificación, en

los que uno de los valores posibles para voltaje corresponde con un valor entre 600 y 750 V. También ocurre en la altura de los andenes, donde uno de los valores posibles fluctúa entre 300 y 380 mm. Estos rangos pueden dar lugar a error, y generan dificultades a la hora de traspassarlo a otros formatos como RailML, donde se contemplan todos los valores que hay entre medias.

- 5) Algunos **nombres de elementos** pueden dar lugar a confusión o malinterpretación, como sucede con los elementos `lineGaugeChanges` y `axleLoadCatChanges`. Por la traducción de ambos al castellano se puede pensar que corresponden a cambios en el ancho de vía (*line gauge*) y en el límite de carga por eje (*axle load*) respectivamente. Sin embargo, hacen referencia al cambio de gálibo y de velocidad según la carga respectivamente
- 6) Sería de gran importancia indicar el **aspecto de las señales**, es decir, qué es lo que indican, ya que el indicar que hay una señal sin saber su significado tiene poco sentido, ya que no se podría valorar si una baliza ha transmitido la información correcta.

### 9.1.2 RailML

RailML supone un sistema bastante completo. En cambio, hay algunos datos que se podrían añadir para facilitar la traducción a los usuarios de UNISIG SS 112:

- 1) En RailML se considera solo el **posicionamiento** puntual, y en algunos casos, la longitud del elemento correspondiente. En cambio, a la hora de determinar un elemento que afecte a un área en el que se incluyen varios tramos, hay que definir el mismo elemento en cada tramo, lo que debe llevar a confusión. Es por esto que se debería incluir algún posicionamiento de área, incluyendo en cada uno de los tramos una referencia al elemento principal al que estén asociados.
- 2) En cuanto a la **direccion** de ciertos elementos, en RailML se definen a veces como direccionamiento estricto, que solo permite uno de los dos sentidos (nominal o inverso). Algunos de estos elementos, en cambio, se pueden definir en UNISIG con ambas direcciones, lo que llevaría a RailML a crear dos elementos para definir uno solo, uno para cada dirección. Por lo tanto, se debería contemplar el restringir menos la velocidad en muchos elementos, como en los cambios de velocidad, de voltaje, de gálibo o de pendiente.

- 3) En las **condiciones de vía**, no se tienen en cuenta algunos de los tipos usados en UNISIG. Estas clases corresponden con las limitaciones en el consumo de corriente y la obligatoriedad de usar la bocina.
- 4) En RailML no se contemplan zonas en las que esté permitido dar **marcha atrás**, mientras que en UNISIG si que se contempla esta opción. Por lo tanto, sería de utilidad incluir un elemento de vía con estas características.
- 5) Existen muchos **dispositivos de señalización** que no están incluidos en RailML como Eurolazos, RIUs y dispositivos y lazos NTC, así como las áreas ETCS.
- 6) Las balizas deberían incluir información acerca de los **telegramas** que pueden transmitir, como especifica UNISIG.
- 7) Tampoco incluye los **valores nacionales** que incluyen diversos límites según regulaciones nacionales.
- 8) Sería importante definir las **rutas** posibles, lo cual no se encuentra ahora disponible, pero se está estudiando su inclusión en RailML 3.0.
- 9) También sería de gran utilidad la reconsideración de restaurar el elemento encargado de las **visualizaciones**, ya que actualmente no hay ningún campo que las tenga en cuenta, y son fundamentales a la hora de realizar las pruebas, para facilitar el diagnóstico de los datos.

## 9.2 CONCLUSIONES GENERALES

A pesar de que ambos formatos parecen tener bastantes carencias, la mayoría de las de RailML se pueden solventar sin realizar ningún cambio, ya que este formato ofrece una gran flexibilidad y permite la inclusión de nuevos elementos y atributos sin ofrecer ningún problema. Los elementos y atributos que sería útil añadir según las propuestas son válidas para concordar con lo especificado únicamente por UNISIG, para facilitar de este modo la búsqueda de equivalencias entre ambos formatos.

Estas propuestas para la extensión de RailML acorde con lo que especifica UNISIG, han sido transmitidas a colaboradores de RailML a partir de este trabajo. Sin embargo, no ha habido respuesta a dichas propuestas todavía, por lo que no se incluyen como resultados reales del proyecto.





La extensión de RailML para UNISIG, incluyendo todos los elementos del SS 112 que RailML no incluye, se especifican mediante esquemas XML en diferentes apartados del documento presente.

Por otra parte, las propuestas indicadas para la mejora del protocolo UNISIG, sí que resaltan sus carencias, ya que este formato no dispone de flexibilidad alguna. Por lo tanto, este formato puede resultar útil para sus usuarios actuales, ya que muchos de ellos han formado parte del desarrollo de dichas especificaciones. En cambio, de cara a su uso como estándar, no tiene ningún futuro en comparación con RailML.

De este modo, se demuestra que RailML es un proyecto con muchas posibilidades de lograr estandarizar el intercambio de información en aplicaciones ferroviarias, contribuyendo así al proyecto VITE y a mejorar la interoperabilidad fundamentalmente entre países europeos, pero también en el resto del mundo. Por lo tanto, este proyecto colaborativo abierto supone un paso más hacia la mejora del sistema ferroviario y de sus posibilidades.

## 10 GLOSARIO DE TÉRMINOS

---

<b>ADIF</b>	Administrador De Infraestructuras Ferroviarias
<b>ATC</b>	Automatic Train Control
<b>ATP</b>	Automatic Train Protection
<b>BTM</b>	Balise Transmission Module
<b>CE</b>	Comisión Europea
<b>CEDEX</b>	Centro de Estudios y Experimentación de Obras Públicas
<b>ERA</b>	European Railway Agency
<b>ERTMS</b>	European Rail Traffic Management System
<b>ETMS</b>	European Traffic Managment System
<b>GSM-R</b>	Global System Mobile – Railway
<b>H2020</b>	Horizonte 2020
<b>I+D</b>	Investigación y Desarrollo
<b>INECO</b>	Ingeniería y economía del transporte S.A
<b>INESS</b>	Integrated European Signaling System
<b>IOP</b>	Interoperability
<b>IP</b>	Innovation Programmes
<b>NTC</b>	National Train Control System
<b>PK</b>	Punto kilométrico
<b>PPP</b>	Public-Private Partnership
<b>RailML</b>	Railway Markup Language
<b>RFI</b>	Rete Ferroviaria Italiana
<b>RIU</b>	Radio Infill Unit
<b>S2R</b>	Shift 2 Rail
<b>S2R JU</b>	Shift 2 Rail Joint Undertaking
<b>SERA</b>	Single European Rail Area
<b>SS</b>	Subset
<b>UE</b>	Unión Europea
<b>VITE</b>	Virtualisation of Testing Environment
<b>WP</b>	Work Package
<b>XML</b>	Extensible Markup Language
<b>XSD</b>	XML Schema Definition

Tabla 19: Glosario de términos

## Referencias bibliográficas

- [1] Shift2Rail Joint Undertaking y Comisión Europea, «Shift2Rail Strategic Master Plan,» Bruselas, 2015.
- [2] ADIF, *Curso sistemas ERTMS. Generalidades*, 2014.
- [3] ADIF y ADIF Alta Velocidad, “Pruebas y ensayos en la infraestructura e instalaciones de ADIF y ADIF Alta Velocidad: Baremo de Precios 2015,” 2015.
- [4] B. Sierra, "Proposal for VITE S2R-OC-IP2-02-2015," Bruselas, 2016.
- [5] «CEDEX,» [En línea]. Available: <http://www.cedex.es>. [Último acceso: 20 Mayo 2017].
- [6] M. Palumbo, «Railway Signalling since the birth to ERTMS,» Noviembre 2013. [En línea]. Available: [http://www.railwaysignalling.eu/wp-content/uploads/2013/11/Railway\\_Signalling\\_since\\_birth\\_to\\_ERTMS.pdf](http://www.railwaysignalling.eu/wp-content/uploads/2013/11/Railway_Signalling_since_birth_to_ERTMS.pdf). [Último acceso: 2 Junio 2017].
- [7] European Comission, «Mobility and transport-History of ERTMS,» [En línea]. Available: [https://ec.europa.eu/transport/modes/rail/ertms/general-information/history\\_ertms\\_it](https://ec.europa.eu/transport/modes/rail/ertms/general-information/history_ertms_it). [Último acceso: 23 Mayo 2017].
- [8] «Página oficial de ADIF,» [En línea]. Available: <http://www.adif.es>. [Último acceso: 25 Mayo 2017].
- [9] UIC, «ERTMS Implementation,» [En línea]. Available: <http://uic.org/ERTMS-Implementation>. [Último acceso: 2 Junio 2017].
- [10] G. M. Linder C., «Datenmodellanalyse zum Austausch von Projektierungsdaten für Stellwerkssysteme in INESS (Analysis of data models for the exchange of project data for interlocking systems in INESS),» 2012.
- [11] VITE, «Lab architecture State of the art analysis,» 2017.
- [12] World Wide Web Consortium, «XML TECHNOLOGY,» [En línea]. Available:

<https://www.w3.org/standards/xml/>. [Último acceso: 3 Junio 2017].

- [13] RailML, «Página oficial de RailML,» [En línea]. Available: <https://www.railml.org>. [Último acceso: 20 Abril 2017].
- [14] UNIFE, «UNISIG: AN INDUSTRIAL CONSORTIUM TO DEVELOP ERTMS/ETCS,» [En línea]. Available: [http://www.ertms.net/wp-content/uploads/2014/09/ERTMS\\_Factsheet\\_8\\_UNISIG.pdf](http://www.ertms.net/wp-content/uploads/2014/09/ERTMS_Factsheet_8_UNISIG.pdf). [Último acceso: 3 Junio 2017].
- [15] UNISIG, «UNISIG Interoperability Test - Guidelines - Subset 110 V3.0.0,» 2011.
- [16] RailML, «Wiki RailML,» [En línea]. Available: <https://wiki.railml.org>. [Último acceso: 28 Mayo 2017].
- [17] UNISIG, «Basics for Interoperability Test Scenario Specifications - Subset 112 V3.6.0,» 2016.
- [18] Agencia Ferroviaria Europea, «Guía para la aplicación de la ETI sobre INF v3.0,» 2015.
- [19] T. Büker, «Railways through Europe,» [En línea]. Available: <http://www.bueker.net>. [Último acceso: 20 Mayo 2017].
- [20] UNISIG, «ERTMS/ETCS System Requirements Specification-Subset 026-7 V3.4.0,» 2014.
- [21] ERA, «CR INF TSI,» 2011. [En línea]. Available: <http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2011:126:FULL&from=EN>. [Último acceso: 23 Mayo 2017].
- [22] European Railway Agency, «ASSIGNMENT OF VALUES TO ETCS VARIABLES,» 2016.
- [23] FEVE, «Reglamento de señales,» [En línea]. Available: [http://www.adif.es/es\\_ES/conoceradif/doc/Feve-RS.pdf](http://www.adif.es/es_ES/conoceradif/doc/Feve-RS.pdf). [Último acceso: 28 Mayo 2017].



- [24] «La terminal de mercancías de Barcelona-Can Tunis,» 2015. [En línea]. Available: <http://www.trenesmania.com/2015/06/la-terminal-de-mercancias-de-barcelona-can-tunis/>. [Último acceso: 1 Junio 2017].
- [25] ADIF, «Sistema europeo de circulacion de trenes,» 2005. [En línea]. Available: [http://www.scf.es/sites/default/files/files/areas/seguridad\\_en\\_la\\_circulacion\\_adif/Manuales\\_de\\_Circulacion/MC-24-SISTEMA\\_EUROPEO\\_DE\\_CIRCULACION\\_TRENES-ERTMS-ETCS.pdf](http://www.scf.es/sites/default/files/files/areas/seguridad_en_la_circulacion_adif/Manuales_de_Circulacion/MC-24-SISTEMA_EUROPEO_DE_CIRCULACION_TRENES-ERTMS-ETCS.pdf). [Último acceso: 1 Junio 2017].